WebSphere® software

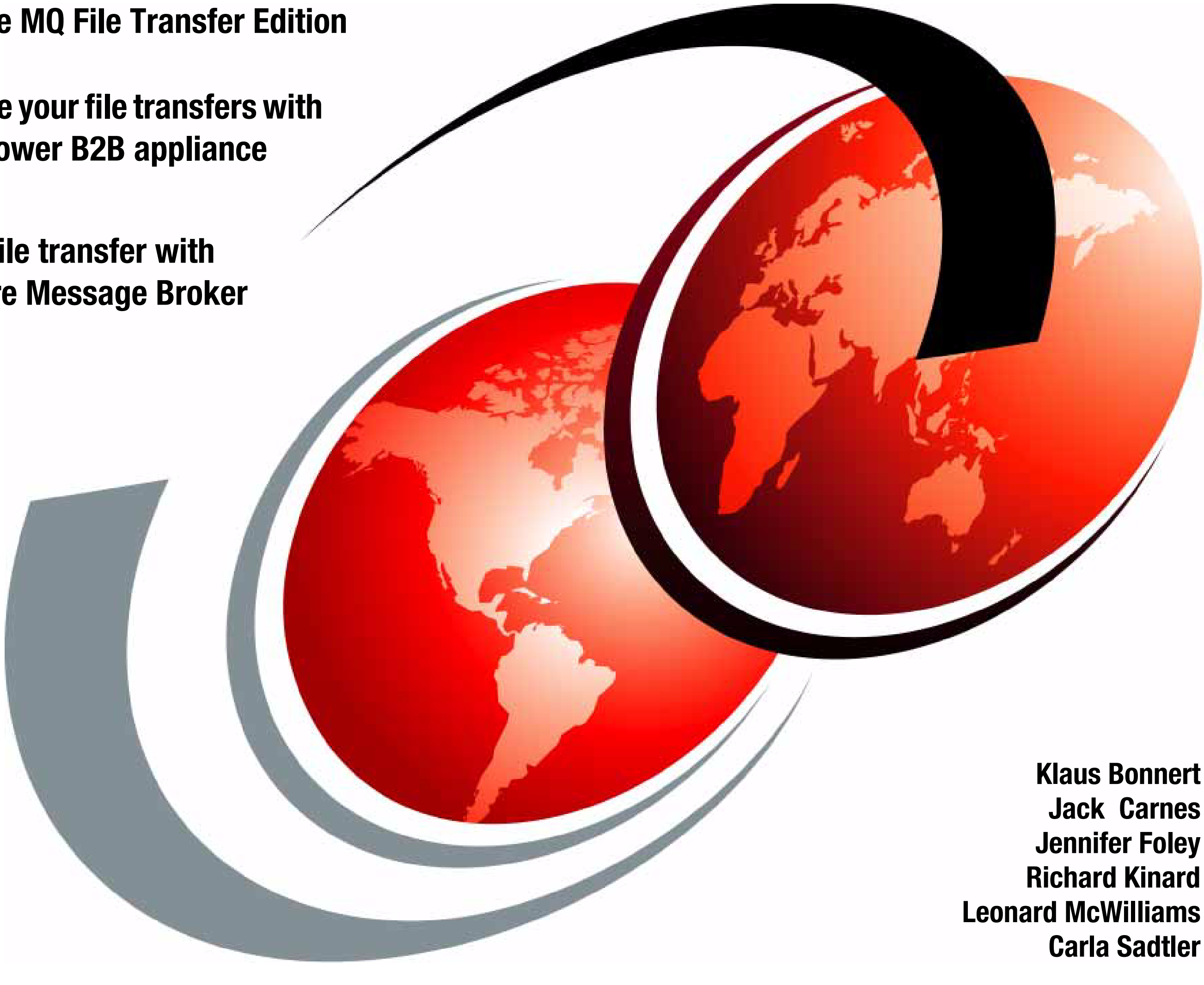
IBM

Multi-Enterprise File Transfer with WebSphere Connectivity

Enable end-to-end file transfers using WebSphere MQ File Transfer Edition

Externalize your file transfers with the DataPower B2B appliance

Enhance file transfer with WebSphere Message Broker



Klaus Bonnert
Jack Carnes
Jennifer Foley
Richard Kinard
Leonard McWilliams
Carla Sadtler

Redbooks

ibm.com/redbooks

International Technical Support Organization

Multi-Enterprise File Transfer with WebSphere Connectivity

September 2010

SG24-7886-00

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (September 2010)

This edition applies to WebSphere MQ File Transfer Edition V7, WebSphere Message Broker V7, and IBM WebSphere DataPower B2B Appliance XB60 with firmware Version 3.8.1.2.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

CICS®
DataPower®
DB2®
IBM®
IMS™
MQSeries®
MVS™
Redbooks®
Redpaper™
Redbooks (logo) ®
System z®
WebSphere®
z/OS®

The following terms are trademarks of other companies:

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redbooks® publication describes how to exchange data between applications running in two separate enterprises reliably and securely. This book includes an overview of the concepts of managed file transfer, the technologies that can be used, and common topologies for file transfer solutions. It then provides four scenarios that address different requirements. These scenarios provide a range of options that can be suited to your individual needs. This book is intended for anyone who needs to design or develop a file transfer solution for his enterprise.

The first scenario shows the use of an HTTPS web gateway to allow files to be transferred from an external web client to an internal WebSphere MQ File Transfer Edition backbone network. This option uses the WebSphere MQ File Transfer Edition Web Gateway SupportPac FO02.

The second scenario uses the WebSphere MQ File Transfer Edition bridge agent to allow files to be transferred from an external File Transfer Protocol (FTP)/Secure File Transfer Protocol (SFTP) server to a WebSphere MQ File Transfer Edition backbone network

The third scenario extends the concept of file transfer between enterprises by introducing more sophisticated transfer capabilities, along with enhanced security. This scenario uses the IBM WebSphere DataPower B2B Appliance XB60 to look at the specific case of file transfers between business partners.

The last scenario also illustrates the integration of the IBM WebSphere DataPower B2B Appliance XB60 and WebSphere MQ File Transfer Edition, but in this case, non-business-to-business protocols are used. The file transfer is further enhanced through the use of WebSphere® Message Broker to mediate the file transfer for routing and protocol transformation within the enterprise.

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

Klaus Bonnert is a Senior Certified IT Specialist based in Stuttgart, Germany. He has been a member of the WebSphere Technical Sales Team in Germany for more than 10 years, working with clients from all industries. His focus is on the integration discipline and the WebSphere Connectivity product portfolio. Prior to working for IBM, Klaus gained comprehensive experience in the IT industry as a Sales Representative and Consultant. He has a degree in Mechanical Engineering from the Technical University in Karlsruhe.

Jack Carnes is an Executive IT Specialist in the WebSphere brand. He has been a member of the WW Connectivity Technical Sales Team for eight years and has over 15 years of experience in messaging and middleware technology. Jack is the world-wide leader for technical sales support and enablement for WebSphere MQ and WebSphere Message Broker for z/OS®. He visits customer around the world and has created numerous training classes for global IBM WebSphere Technical Specialists. He is an IBM Senior Certified IT Specialist and Master Certified by the Open Group. He is a member of the Connectivity Competency Leadership Team responsible for setting the direction of the Connectivity Technical Specialists.

Jennifer Foley is a WebSphere on System z® IT Specialist based in Dallas, TX. She began her career with IBM as a college intern in IBM Retail Store Solutions working on a Linux®-based operating system for retailers. Since joining IBM full time, she has been working with customers to grow and modernize their WebSphere portfolio on System z. Her current product expertise is in the WebSphere Connectivity and Application Infrastructure portfolios. She has developed customer demos available through the IBM DEMOcentral organization and written technical documents for internal use within IBM. She has a Bachelor of Science degree from the University of Oklahoma in Computer Science with a minor in Mathematics.

Richard Kinard is the Product Manager for WebSphere DataPower® Appliances. He is a subject matter expert in business-to-business technologies and has over 12 years of experience designing, developing, and implementing business-to-business solutions. He has worked on many initiatives with Internet standards organizations to promote business-to-business interoperability and was a Senior Product Manager of a successful business-to-business application prior to working for IBM.

Leonard McWilliams is a Consulting IT Specialist on the WebSphere Federal Sales Team working primarily with WebSphere MQ and WebSphere Message Broker in classified accounts. He has 35 years of IT industry consulting and application development experience, which includes messaging, database management, and geographical information systems. He has bachelor's degrees in Biological Science, Philosophy, and Music from the University of Kansas and a Master of Education degree from Antioch University.

Carla Sadtler is a Consulting IT Specialist at the ITSO, Raleigh Center. She writes extensively about WebSphere products and solutions. Before joining the ITSO in 1985, Carla worked in the Raleigh branch office as a Program Support Representative, supporting MVS™ customers. She holds a degree in mathematics from the University of North Carolina at Greensboro.

The authors (left to right): Richard Kinard, Jack Carnes, Jennifer Foley, Leonard McWilliams, Klaus Bonnert

Thanks to the following people for their contributions to this project:

Margaret Ticknor
International Technical Support Organization, Raleigh Center

Ben Mann
IBM UK

Adrian Preston
IBM UK

T. Robert Wyatt
IBM US

Robert Simons
IBM US

Elliott Gregory
IBM UK

Christopher Harris
IBM UK

Dominic Evans
IBM UK

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author - all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBM-Redbooks
- Follow us on twitter:

  http://twitter.com/ibmredbooks
- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806
- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm
- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Part 1

Multi-enterprise file transfer concepts

This part discusses the concepts, runtime topologies, and IBM products used to build multi-enterprise file transfer solutions. This part contains the following chapters:

- Chapter 1, “File transfer concepts, technologies, and best practices” on page 3
- Chapter 2, “Multi-enterprise file transfer topologies” on page 15
- Chapter 3, “Product overview” on page 21

1

File transfer concepts, technologies, and best practices

This chapter presents an overview of how businesses are using file transfer protocols to move files both internally and externally and also provides a brief historical view of file transfer technologies. This chapter compares internal file transfer to external enterprise file transfers by examining the similarities and differences that make them separate technologies.

We also discuss the challenges surrounding the multi-enterprise use of the File Transfer Protocol (FTP), how managed file transfer can overcome these issues, and what best practices businesses should consider when looking to move files within and out of their enterprise.

1.1 Introduction

For many organizations, the exchange of files between business systems remains a common and important integration methodology. Files are the simplest unit of data to exchange and often represent the lowest common denominator for an enterprise infrastructure.

Although the exchange of files is conceptually simple, doing so in the enterprise is a challenge to manage and audit. This difficulty is brought into clear focus when an organization needs to perform file transfer with another business organization, perhaps using a different physical network, with different security requirements, and perhaps a different governance or regulatory framework.

Despite an abundance of new technologies that include web services, Web 2.0, and many robust existing technologies such as Enterprise Messaging, file transfer remains a common method of integrating business systems.

1.1.1 Basic FTP

File transfer has a long history. There are many existing tools that support it in some form. The simplest and best known technology for file transfer is the FTP, which was first made available in UNIX® systems in the 1970s. Today, the broad availability of FTP on almost all platforms makes it an easy choice when the need to exchange files arises. However, performing mission-critical file transfers using FTP does have issues with limited reliability, recoverability, security, and auditability.

1.1.2 Managed file transfer

Managed file transfer addresses the needs that organizations have to configure, track, and audit file transfer activity consistently. Typically, an organization that uses managed file transfer has the following needs:

- Auditability: File transfer activity must be logged so that administrators can determine where each file is sent and when the transfer occurred. The transfer log needs to be centrally accessible.
- Security: File transfer requests must be accepted only from authorized people or application systems.
- Recoverability and reliability: Network or other errors that might interrupt a transfer must not cause the transfer to be abandoned or partial files to be received.
- Platform connectivity: File transfers must span multiple platforms.

1.2 Brief history and challenges of file transfer

The most commonly known network protocols are Transmission Control Protocol and the Internet Protocol (TCP/IP), which were the first two networking protocols defined to the Internet Protocol Suite. The TCP/IP model comprises four layers:

- Application
- Transport
- Internet
- Link

As the distributed computing model grew, the enterprise use of TCP/IP grew to support the local area networks and the first ventures into Internet computing.

The FTP was first introduced in 1972. Since then, FTP has been helping companies move volumes of batched and single files between distributed servers. As other communication protocols have been introduced, many protocols for moving files have emerged. Other application layer technologies, such as Hypertext Transfer Protocol (HTTP), HTTP over Secure Sockets Layer (SSL) (HTTPS), Simple Asynchronous File Transfer (SAFT), Secure Copy (SCP), Secure File Transfer Protocol (SFTP), and File Transfer Protocol over SSL (FTPS), have been adopted to meet business demands that FTP alone cannot.

Enterprises today depend on a mix of technologies to move files across their internal systems. These technologies include home-grown solutions, often built around FTP and vendor solutions. The vendor solutions are typically managed file transfer solutions that provide enterprises with features to secure, configure, track, and audit file transfer activity consistently.

Many customers move to a managed file transfer solution to satisfy regulatory-mandated compliance requirements. Compliance mandates, such as Health Insurance Portability and Accountability Act (HIPAA), Payment Card Industry Data Security Standard (PCI DSS), Voluntary Product Accessibility Template (VPAT), and Sarbanes-Oxley Act (SOX), require that the entire transaction flow be secured, auditable, documented, and accountable. As a result, companies must address the inherent weaknesses that exist with basic File Transfer Protocols.

Moving files in and out of protected networks also creates many security risks that cannot be addressed with just basic file transfer. Entities typically choose to exchange data with external partners through one of the application layer protocols (for example, SFTP and HTTPS), proprietary protocols, email, or through business-to-business technology and standards. Most commonly, organizations use a hybrid of technologies to move data in and out of their protected networks.

The most secured and reliable way to move files between organizations is through business-to-business messaging standards like EDIINT AS1, AS2, and AS3, or ebMS, which are specifically designed for securely exchanging data over a public network. Business-to-business applications seek to improve organizational partnerships and transform the partnerships into inter-organizational relationships by acknowledging that the trading entities are known to one another and all users are registered. The data exchanged allows for organizations to directly exchange information in a secure, standard method.

Business-to-business messaging protocols are standards that utilize application layer protocols and provide mechanisms for securing the data through encryption, signatures, and non-repudiation of sender and acknowledgment. The protocols are typically a wrapper or envelope that encompasses the business-to-business document or payload.

Business-to-business document standards like EDIX12, EDIFACT, HIPAA, HL7, and ebXML are designed to be cross-industry standards that provide a single architecture that utilizes a common uniform data format for electronic communications.

Typically, business-to-business solutions encompass traditional File Transfer Protocols, but also include the ability to move files according to the published business-to-business messaging and document standards. Additonally, vender business-to-business offerings generally include partner profile management and transaction-viewing capabilities.

As the dependency on FTP and similar technologies and the need to externalize file transfers has grown, the limitations of the FTP and other application layer protocols has become a challenge for many companies. While companies have grown their file transfer infrastructure using these application layer protocols, the lack of security and management capabilities have sent them looking for better solutions for security and management of their file transfers.

Challenges surrounding FTP in a multi-enterprise file transfer

File transfer is the simplest form of exchanging data between business entities and requires a common integration. The lowest common denominator to move this file is typically FTP.

FTP is a standard network protocol used over a TCP/IP network that allows businesses to move files between disparate systems regardless of operating systems. This flexibility is largely due to the client-server architecture of FTP. The inclusion of FTP into almost all operating systems has enabled its widespread use. This pervasive use of the technology has presented businesses with many challenges regarding the use of FTP. These challenges include but are not limited to the reliability, security, auditing/visibility, flexibility, and operational costs surrounding the use of FTP.

Limited reliability

Lacking checkpoint restart capability, file delivery is unreliable when a network interruption occurs. This often results in corrupt or partial files at the destination. Additionally, even with a successful transfer, the data can still be unusable at the destination due to a lack of character-set conversion. Ultimately, this can be costly for businesses that rely on FTP to move mission-critical files. Many companies find themselves with staff dedicated to cleaning up incomplete or failed transfers.

Limited security

FTP often requires a user name and password to send a file. These user names and passwords are sent across with the file as plain text. Additionally, many implementations of FTP do not offer privacy, authentication, or encryption. With checksum not available for all implementations of FTP, it can be almost impossible for companies to know or be able to show that the data that was supposed to be sent was actually sent without being modified.

Many implementations of FTP also lack a non-repudiation capability that allows for businesses to receive digital acknowledgement with trading partners that they successfully received the transfer. (*Non-repudiation* is the concept that an organization cannot refute the validity of a statement or contract). Non-repudiation typically must be in place to meet government standards regarding transfers in a court of law.

Limited auditability and visibility

With no centralized monitoring or management, FTP transfers can be nearly impossible to track from start to finish as the file makes hops across the enterprise. Logging capabilities are often limited and might only record transfers between systems that are directly connected. The limited logging requires companies to check logs at the source and destination servers for every hop that a file makes. This can make it difficult to track a file across machines. Often, this might not even allow for a complete picture of where the file has traveled.

Limited flexibility

Single-threaded, FTP can only send or receive a file one at a time. To initiate a transfer, FTP also requires both the sending and the destination machines to be available simultaneously. FTP only allows for point-to-point transfers. This can mean that systems not directly connected together might have to be routed administratively through other boxes. FTP also lacks the ability to allow companies to prioritize the transfers.

These flexibility limitations often make it difficult or impossible for companies to automate their file transfer processes. When automation is possible, any scripts used during the transfer process generally reside on the machine on which they are used. This can require changes to be made on various servers that require platform-specific skills to make the modifications.

Often, these limitations paired together dictate what and how companies can move data across and outside their enterprises.

High maintenance costs

Seemingly free, FTP can be costly for companies as they try to develop around the inefficiencies. Requiring companies to dedicate resources to this task, the efforts to build, maintain, and support a custom solution built around FTP can quickly add up, causing companies to spend unexpected funding on this free File Transfer Protocol.

1.3 Overcoming FTP challenges with managed file transfer

Even with the availability of newer technologies such as web services and Web 2.0, file transfer still remains one of the most common ways for enterprises to exchange data. As more data is exchanged internally and externally, the need for more efficient means of moving files has become prevalent. This has brought about the concept of managed file transfer. Typically, managed file transfer refers to software solutions that allow for secure transfer of data from one location to another. A managed file transfer system introduces control, management, and auditability to address problems that arise when file transfers are used to integrate or connect business systems in the organization. Generally, managed file transfer solutions have features such as reporting the completion status of file transfers, auditing, global visibility, automation, and non-repudiation. These features are specifically designed to overcome the common challenges surrounding the enterprise use of FTP.

The following sections discuss how managed file transfer overcomes the current challenges of reliability, security, availability, flexibility, and costs when using FTP.

1.3.1 Improved reliability

Managed file transfer software tries to ensure that file contents only appear at the destination completely intact. The techniques for assured delivery vary among managed file transfer solutions, but most include the ability to resume or restart a file transfer that is interrupted because of network or system availability. Many also include the ability to perform common code character set conversions based on selections specified when initiating a file transfer that enables the software to detect operating systems.

1.3.2 Improved security

Allowing for encryption of data, managed file transfer software keeps user IDs and passwords secured while in flight. Many offerings of managed file transfer software also include a checksum feature that allows a business to guarantee that the data being transferred is in its original form and has not been corrupted or tampered with.

Additionally, managed file transfer software allows companies to utilize non-repudiation that allows trading partners to be notified when the transfer has been received successfully, subsequently reducing the risk for conflicts or litigation. For file transfer, namely external file transfers, an organization cannot dispute that they received a file whenever a message disposition notification is given by specific protocols saying that the transfer is complete. This message disposition notification is a digital signature that the receiver did receive the transfer.

1.3.3 Improved auditability and visibility

Complete and detailed audit logs of the entire journey that a file takes are one of the main features typically offered in managed file transfer software. The logs typically show where the file originated, where the file went, who sent it, who initiated the transfer, who received the file, when the transfer was initiated, and when it was completed. Additionally, many of the software offerings for managed file transfer include the capability to control and monitor file transfers from a central location. The centralized control allows for administrators to easily keep an eye on where things are flowing throughout the enterprise.

1.3.4 Improved flexibility

Managed file transfer software strives to allow for time-independent transfers. Several of the most robust managed file transfer offerings include features that allow companies to initiate file transfers independent of source and target systems being available. Many of these managed file transfer offerings allow companies to send files to systems not directly connected, without requiring additional scripting or manual work. Furthermore, many managed file transfer solutions allow for files of any size to be sent across their technology. This allows the company to function based on business demands instead of technology dependencies.

Many solutions for managed file transfer include the ability to reconfigure and deploy a file transfer instantaneously from anywhere in the infrastructure. Moreover, many offerings of managed file transfer allow for multi-threaded transfers that enable businesses to send and receive multiple files at the same time.

The centralized control and monitoring available in many of the managed file transfer offerings also allows for automation to be set up in a central location. Many offerings expand the abilities for automation by integrating or building on scripting languages. Many of the offerings for managed file transfer include automation features for scheduling, triggering, and event-driven transfers.

1.3.5 Cost effectiveness

Companies looking to reduce costs can capitalize on the industry knowledge and experience of managed file transfer vendors by utilizing a managed file transfer solution. By utilizing offerings from managed file transfer solutions, companies can eliminate the costs associated with continually updating, maintaining, and improving their home-grown file transfer solution. Additionally, by utilizing the managed file transfer implementations, they will be able to take advantage of the continual improvements to features as business demands change and advance. Ultimately, this can allow for a higher utilization of the technology, while giving companies significant cost-savings.

1.4 Managed file transfer best practices

In this section we review best practices for file transfers that span both internal and external networks.

1.4.1 Internal managed file transfer best practices

Typically, companies are unaware of the various file transfers taking place in their protected network. It is common to walk into a business and find a variation of the scenario shown in Figure 1-1.

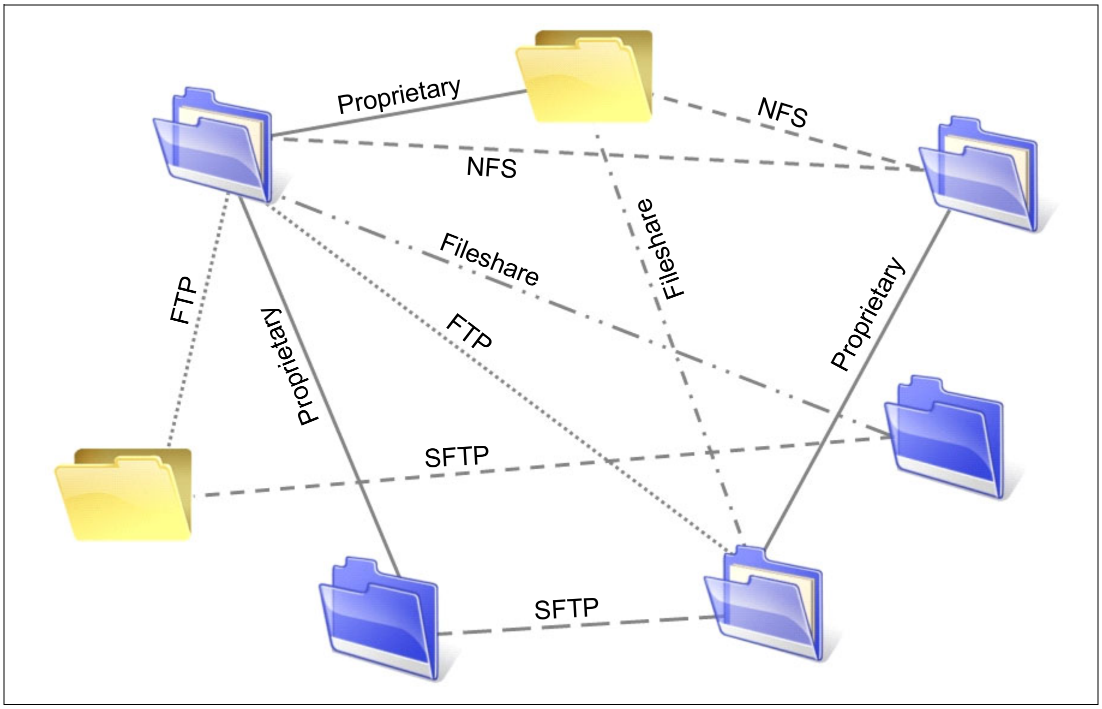


Figure 1-1 Typical file transfer topology found within companies

The combinations of many protocols and technologies leave your technicians scrambling to maintain disparate systems. Another challenge facing organizations, keeping employees' skills for the various protocols up to date, can be challenging too. Additionally, the more technologies that you add to an organization, the harder it is to control who is using the technologies and to monitor the activity.

Internal managed file transfers re-enforce a company's service-oriented architecture (SOA) strategy. Ideally, this strategy is built on top of a messaging backbone to reduce redundant technology infrastructures, administrative overhead, and skills required of employees. The ideal managed file transfer architecture contains automation, centralized and event-based logging, centralized monitoring, centralized setup and management, and a documented and standardized transport. The infrastructure contains:

- The ability to secure files in transit
- Flexibility in file transfers
- Granular user control over file transfers
- Monitoring of a file's journey
- Auditing of transfers
- Visibility of transfers
- Checkpoint restart of transfers

The ideal internal managed file transfer topology utilizes a single reliable transport (Figure 1-2). The topology cuts down on the cost and time for information technology (IT) development and maintenance by eliminating the need to write code. This allows for the configuration and extension of the managed file transfer software to consolidate IT administration and operational efforts. The ideal managed file transfer topology preserves the integrity of data to support a company's compliance requirements by being secure, reliable, and resilient and allowing auditing. The topology for moving files easily integrates with a SOA enterprise service bus to transform, parse, and route data.

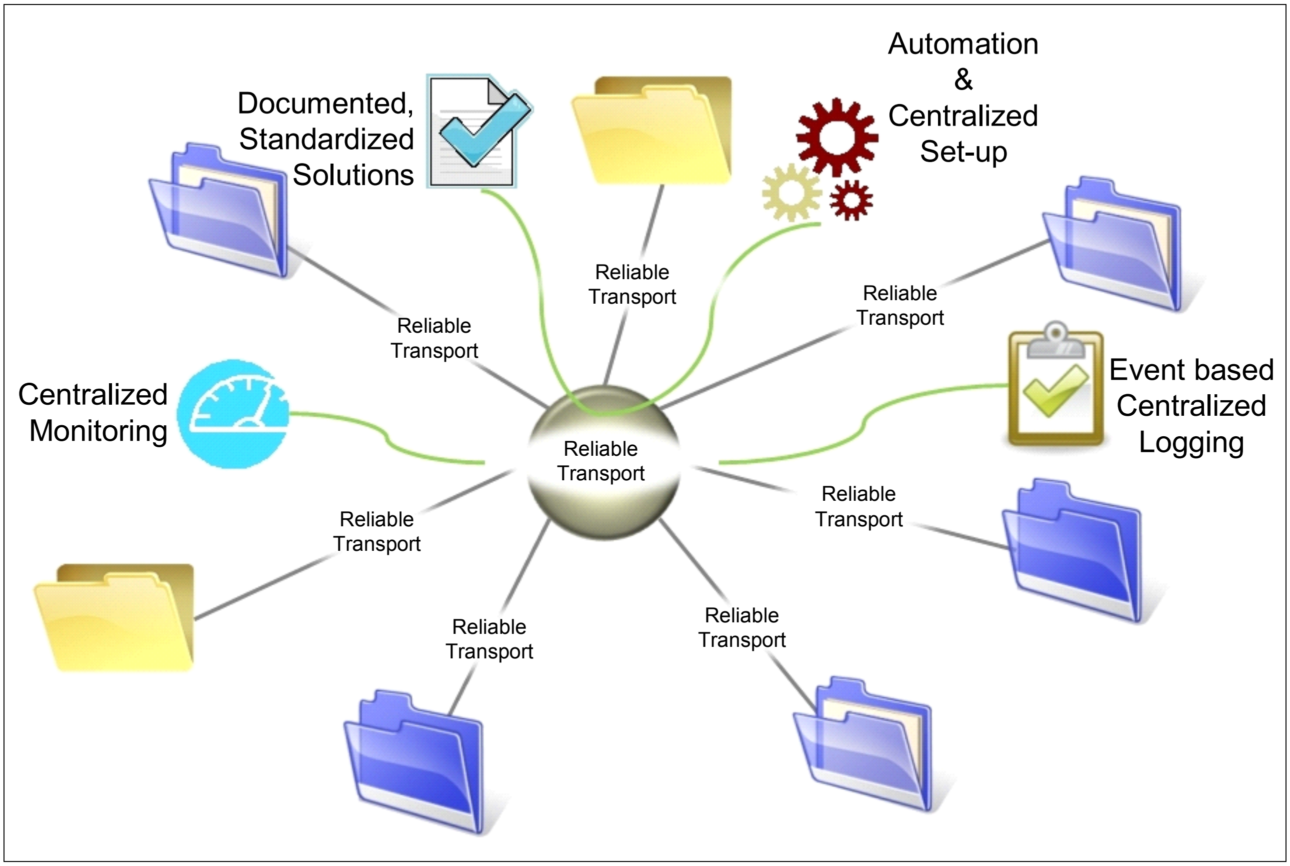


Figure 1-2 Ideal managed file transfer topology

The key features for internal managed file transfer best practices are:

- Common reliable transport protocol
- Centralized monitoring
- Event-based, centralized audit logging
- Automation
- Centralized setup and configuration
- Documented, standardized solutions
- Check-point recovery
- Centralized management

1.4.2 External multi-enterprise managed file transfer best practices

Enterprises working with various trading partners typically are sending data across multiple firewalls and are working with multiple document, messaging, and transport standards. The various protocols and document standards require proper management and support. As more trading partners are integrated into the managed file transfer topology, it becomes necessary to manage trading partner IDs, authenticate users, and authenticate the data flowing into and out of the company.

The ideal multi-enterprise managed file transfer topology places a gateway between a company's internal managed file transfer topology and its trading partner. Typically, this gateway resides in the demilitarized zone (DMZ). The gateway should be capable of handling a wide range of protocols to meet current and future business requirements. Additionally, the gateway should be designed to handle large amounts of file transfers, meet or exceed performance expectations, and meet security certification standards.

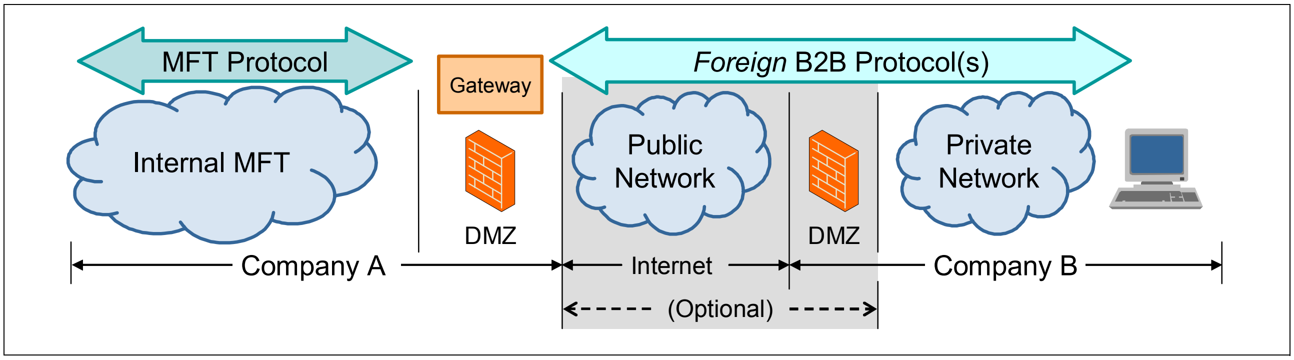


Figure 1-3 Ideal External multi-enterprise managed file transfer topology

The gateway should also support standards-based business-to-business protocols and be designed specifically for DMZ deployments. B2B gateways allow for ease of managing and connecting to trading partners using industry standards and provides trading partner management for business-to-business governance, utilizes business-to-business protocol policy enforcement, implements access control, executes message filtering, and performs data security. A trait also common to most B2B gateway products is a user interface for B2B configuration and transaction viewing. The user interface of the B2B gateway is used to correlate documents and acknowledgements while displaying all associated events.

The key features for external multi-enterprise managed file transfer best practices are:

- Trading partner management
- Hardened security for DMZ deployment
- Business-to-business governance and security
- Broad range of business-to-business and transport protocol support
- User interface for configuration and transaction viewing
- Interface for trading partner transaction viewing
- Assured delivery with automatic resend

1.5 Comparing internal and external file transfers

Any time that a file moves there are certain basic concerns regarding the movement between the source and the destination. The severity and detail of these concerns vary depending on whether the file in transit is subject to meeting industry compliance standards, moving over a public network (Internet), or staying inside the protected network. The level of risk associated with moving a document outside the enterprise is the key differentiator between internal and external file transfer and drives the types of technologies and products used to mitigate that risk.

1.5.1 Control

The amount of control that one enterprise has over file transfers varies depending on whether the files are transferred within the enterprise or outside the enterprise to external partners.

Internal file transfers

File transfers moving only inside an organization can be controlled easier than an external file transfer. While moving a file internally, entities can control the coordination of the source and destination targets, view logging data available at the source or the destination, and store information pertaining to the transfer that is required for auditing. These activities can be performed with a complete picture of the components involved in the transfer.

The level and type of information included in logging varies depending on the protocol chosen for the file transfer. Additionally, with access to both source and destination targets, organizations can see a complete picture of the file transfer and monitor the source or destination targets as needed to alert them of issues relating to the file transfer.

External file transfers

Moving files outside of an organization's firewalls forces the organization to relinquish a certain amount of control over the transmission. With the inability to control what is going on at the external end of the transfer, coordinating the submission of the transfer can be challenging. Many of the protocols require the sending and receiving platforms to be available at the time of the transfer initiation. This can require an organization to resubmit a transfer request multiple times before the transfer completes. Additionally, many of the protocols can report a successful transmission even when the file did not actually transmit for various reasons. Without the ability to see the transmission completely, organizations might have no idea whether a file was successfully received if they use traditional application layer file transfer protocols. Standards such as AS2 help resolve this issue through a Message Disposition Notification (MDN) that provides a message-level acknowledgment that the transfer was a success.

Monitoring and complete visibility can be a challenge with external transfers. Most of the typical protocols do not provide any visibility or monitoring into the transmission. Certain protocols may log at the source, destination, or both, but without access to both logs, entities are left with an incomplete picture.

1.5.2 Authentication and data validation

When receiving a data transmission, whether internally or externally, there are questions regarding the transmission. Namely, who sent it and is this the data that should have been sent. The requirements for verifying the sender and the validity of the data vary depending on whether the transfer occurred internally or externally.

Internal file transfers

In internal file transfers, organizations typically do little user authentication or data validation. Typically, the sender's user ID is authenticated against the destination operating system to ensure that the sender has access to the system and permission to access or write to the desired destination directory. This process does not usually include any data validation to ensure that the data is safe because the electronic transmission is coming from a trusted source inside the protected network.

External file transfers

User ID authentication and data validation are a primary concern for external file transfers. However, the authentication concerns begin before a user ID or the data comes into view. First, an external file transfer should verify that the Internet Protocol (IP) address or range of addresses being used in the transmission is allowed access to the systems inside the DMZ. This is typically done at the external firewall utilizing inbound firewall rules. Once the IP address or range of addresses is determined to be valid, the user or partner ID involved in the transmission should be authenticated before files can be exchanged. Once the user or partner IDs are validated, also validate the data should to ensure that the file type and format is allowed by the receiving system.

1.5.3 Network security

As files travel across the network, security measures should be put in place to protect the data as it is transmitted.

Internal file transfers

Typically, encryption inside a protected network is not an issue for organizations. The exception to this rule occurs when the data in question is subject to compliance standards, such as Payment Card Industry (PCI). These standards can often require data to be encrypted when on a file system and in flight. When this is necessary, entities look to encrypting the data or encrypting the file system and the transmission channel. Additionally, there are few, if any, concerns about ports being used for the electronic communication.

External file transfers

Every open port through a firewall is another possible entry into the organization's demilitarized zone (DMZ), protected network, or both. External file transfers might require one or more ports to be opened through the external firewall depending on the protocol in use. The inner firewall needs to be locked down to allow ports only open from the gateway sitting in the DMZ. As more protocols are added to the organization, more ports need to be opened through the firewalls. Proper network and data security must be used together to ensure that the data and the internal protected network are protected.

Once the file leaves an organization's secured zone, if the data contains sensitive information, such as user credentials or account numbers, it immediately becomes a security risk. The risk typically requires the data to be encrypted while it is in the DMZ and while in flight into and out of the DMZ to and from an external partner.

2

Multi-enterprise file transfer topologies

In this chapter, we introduce multi-enterprise file transfer topologies utilizing a variety of IBM products inside the protected network, all attached to a MQ backbone that acts as a conduit between each system and application. These topologies utilize WebSphere MQ File Transfer Edition to move files between each of the systems and applications.

The topologies extend and build on the basic single queue manager WebSphere MQ File Transfer Edition topologies that were introduced in *Getting Started with WebSphere MQ File Transfer Edition V7*, SG24-7760. Any of the WebSphere MQ File Transfer Edition topologies depicted in the above-listed book work with multi-enterprise file transfers.

2.1 Introduction

For the purpose of this book, end-to-end file transfer is defined as multi-enterprise file transfer where data is transferred between two or more companies, typically over the Internet, but the data can also be exchanged over a Value Added Network (VAN) or even point-to-point over a direct connection into each company's network. In many cases the files originate from a system in one enterprise and are destined for a system in another enterprise. These files sometimes need to be routed based on content or file type and also need to be transformed into a file format specific to the receiving company's systems.

Multi-enterprise file transfer consists of a minimum of two companies, and as such, you will not always know what method of internal file processing is done by the external company. It might use similar products as you or it might use a custom or proprietary solution to process and move files internally. For this reason, it is difficult to gain end-to-end visibility of the entire transaction flow unless both companies use the same technology or have a mechanism for providing a standard way of viewing transaction state end-to-end. With this in mind, this chapter depicts topologies that use IBM products on only the internal company side of the external partner relationship, whereas the external company side is a high-level topology.

Multi-enterprise file transfer is typically broken down into two primary methods of transferring files:

- File transfers over application layer protocols, such as:
  - File Transfer Protocol (FTP)
  - Secure File Transfer Protocol (SFTP)
  - Hypertext Transfer Protocol (HTTP)
  - Simple Mail Transfer Protocol (SMTP)/POP
- Business-to-business-enabled managed file transfer using business-to-business messaging protocols, such as:
  - EDIINT
    - AS1
    - AS2
    - AS3
  - ebXML
  - RosettaNet
  - SOAP with Attachments

Figure 2-1 illustrates a high-level file transfer topology. The remainder of this chapter provides detailed topologies for the internal systems side (the top) of Figure 2-1.

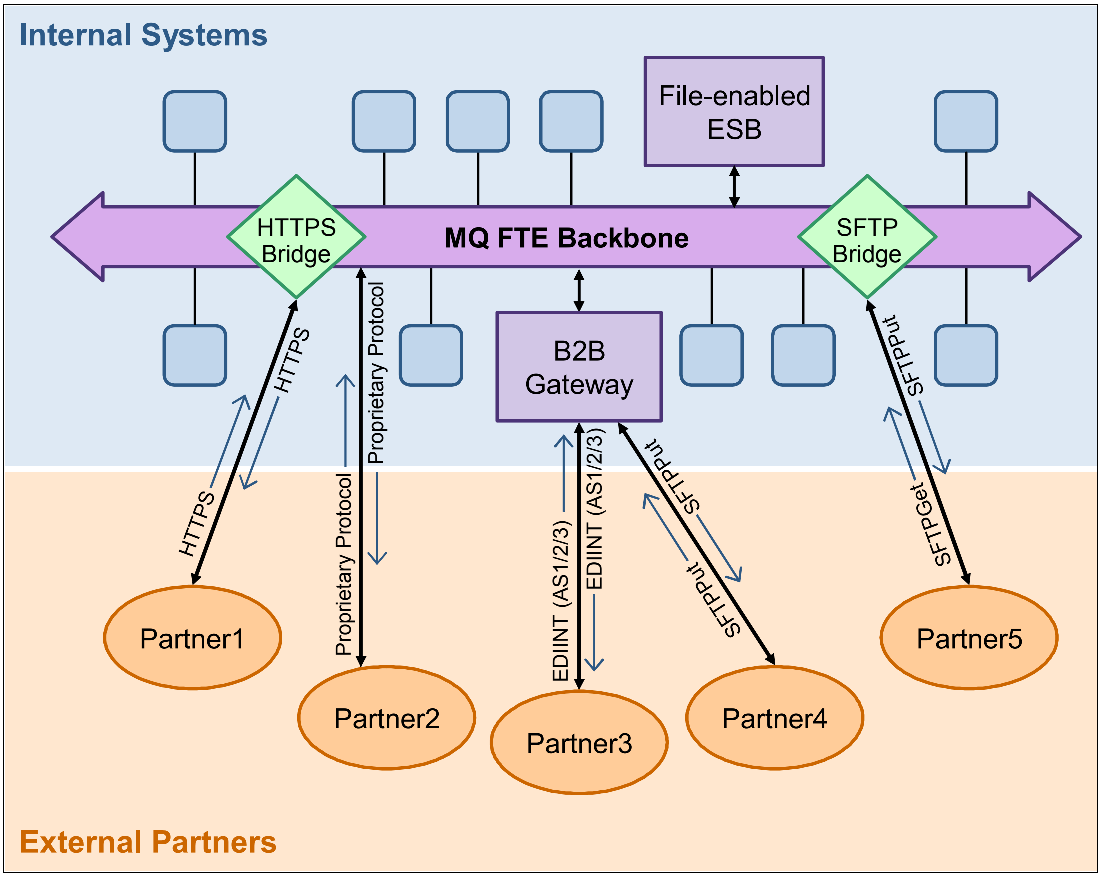


Figure 2-1 High-level file transfer topology

2.2 Application layer protocol topology

The application layer protocol topology consists of all protocols and methods for process-to-process communications over an Internet Protocol (IP) network using transport layer protocols to establish connections between hosts. The most common application layer protocols used for transferring files over the Internet are FTP and SMTP/POP. However, SFTP and HTTP are gaining significant adoption as well. In many cases, the data to be transferred over these protocols is not encrypted. Many companies prefer to use a transport layer security protocol like Transport Layer Security (TLS) or Secure Socket Layer (SSL) in conjunction with FTP, HTTP, and SMTP/POP to encrypt segments of the network connection. The topologies depicted here utilize transport layer security as a best practice.

A technology not depicted here is proprietary third-party file transfer technology that utilizes proprietary application layer protocols built on top of TCP/IP. These protocols include Sterling Commerce's Connect:Direct protocol, RepliWeb's RMFT protocol, Microsoft®'s Media Transfer protocol, and many more. Typically, the proprietary technology is client/server based, where the client resides on the external company's systems and the server resides in the internal company's DMZ.

Topology overview

Figure 2-2 shows a typical topology for file transfer over the application topology layer, using IBM products. The topology consists of two network zones.The demilitarized zone sits between two firewalls and contains systems that can be accessed by outside entities over the Internet. These systems, which typically include edge servers, IBM DataPower Appliances, or both, act as a buffer to protect the applications and data stores that need to be secured in the protected network zone.

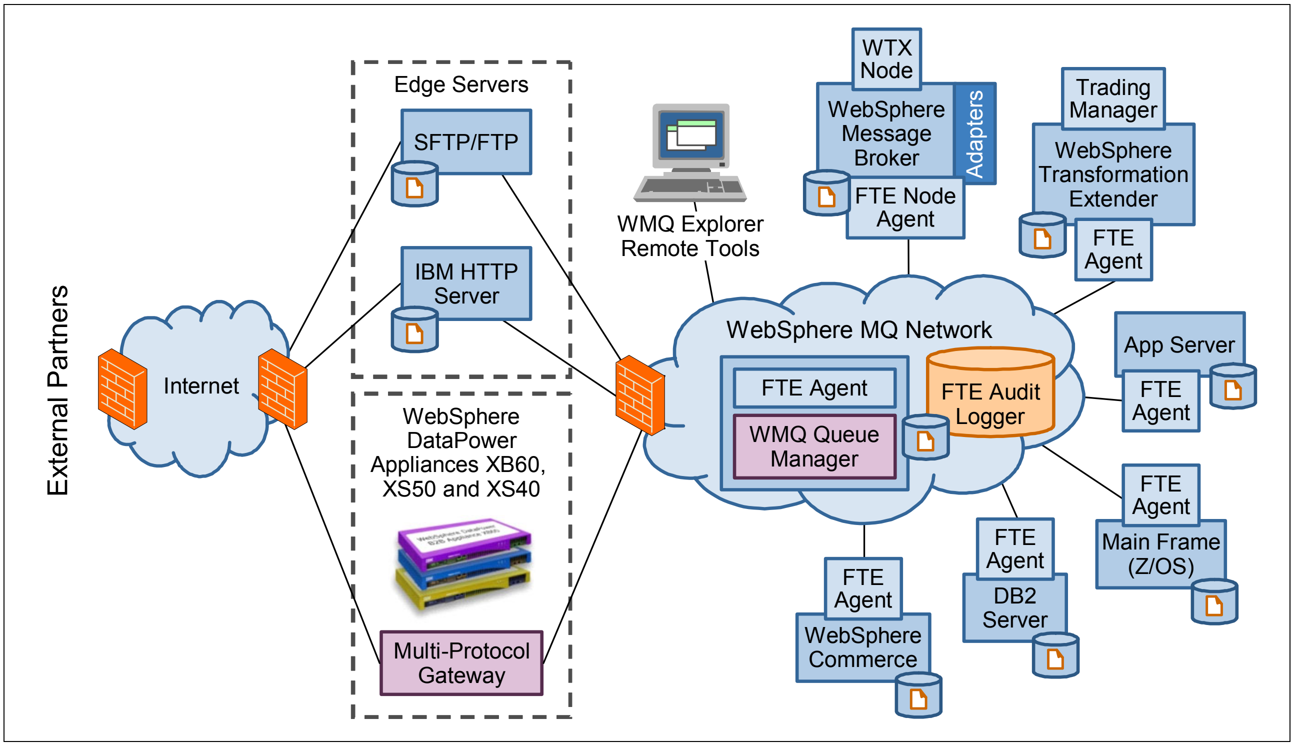


Figure 2-2 Application layer protocol topology with IBM products

Demilitarized zone systems

The systems in the demilitarized zone provide the ability to connect to external partners utilizing FTP, SFTP, and HTTP servers or services. Systems in the DMZ are typically used as intermediaries for requests from clients. They forward the requests or files to and from servers inside the protected network zone. These systems are commonly referred to as edge servers and can be proxies, web servers, messaging servers, proprietary connectors, and even email servers. Note, however, that email server architecture significantly differs from the architecture represented here and is not discussed because it is not our primary focus of this book.

DataPower appliances are a prime example of devices that can be deployed in the demilitarized zone to provide edge security, threat protection, authentication, authorization and auditing of connected partners, integration to popular virus detection applications, and more. These network devices are purpose-built and easily dropped into existing network environments. They natively accept requests over multiple protocols, including FTP, SFTP, HTTP, and many more. The appliances allow protocol bridging from input to output, providing you with the ability to consolidate many edge type servers into a single device.

Protected network systems

The systems in the protected network are typically systems that are responsible for file processing, for example:

- Transformation with applications such as WebSphere Transformation Extender
- Business process management with applications such as WebSphere Process Server
- Application integration with enterprise service bus applications such as:
  - WebSphere Message Broker
  - WebSphere Enterprise
  - Service Bus
  - DataPower XI50 Appliances

The protected network also contains applications like enterprise resource planning applications, customer relationship management applications, inventory management applications, and custom-developed applications. All of these applications and systems typically need a data or message store for file and transaction persistence (often in the form of hard drive subsystems), storage area networks, databases, and message queues. Systems inside the protected network are all attached to a WebSphere MQ network, allowing each system to interact with each other on a single common data bus.

2.3 Business-to-business-enabled managed file transfer topology

The business-to-business enabled file transfer protocols consist of a group of standards that wrap or encompass the payload to be transferred. The wrapper is commonly referred to as a messaging envelope and defines attributes such as sender ID, receiver ID, message ID, data security (encryption and signatures), and message-level acknowledgement requests. The most widely adopted business-to-business messaging protocols are EDIINT AS1, AS2, and AS3. However, ebXML and SOAP with Attachments are gaining significant adoption as well.

Topology overview

The topology depicted in Figure 2-3 also consists of two network zones:

- The demilitarized zone
- The protected network

The main difference in this topology is that the IBM WebSphere DataPower B2B Appliance XB60 is used in the DMZ instead of the traditional edge type of servers. There are no changes in the protected network zone topology.

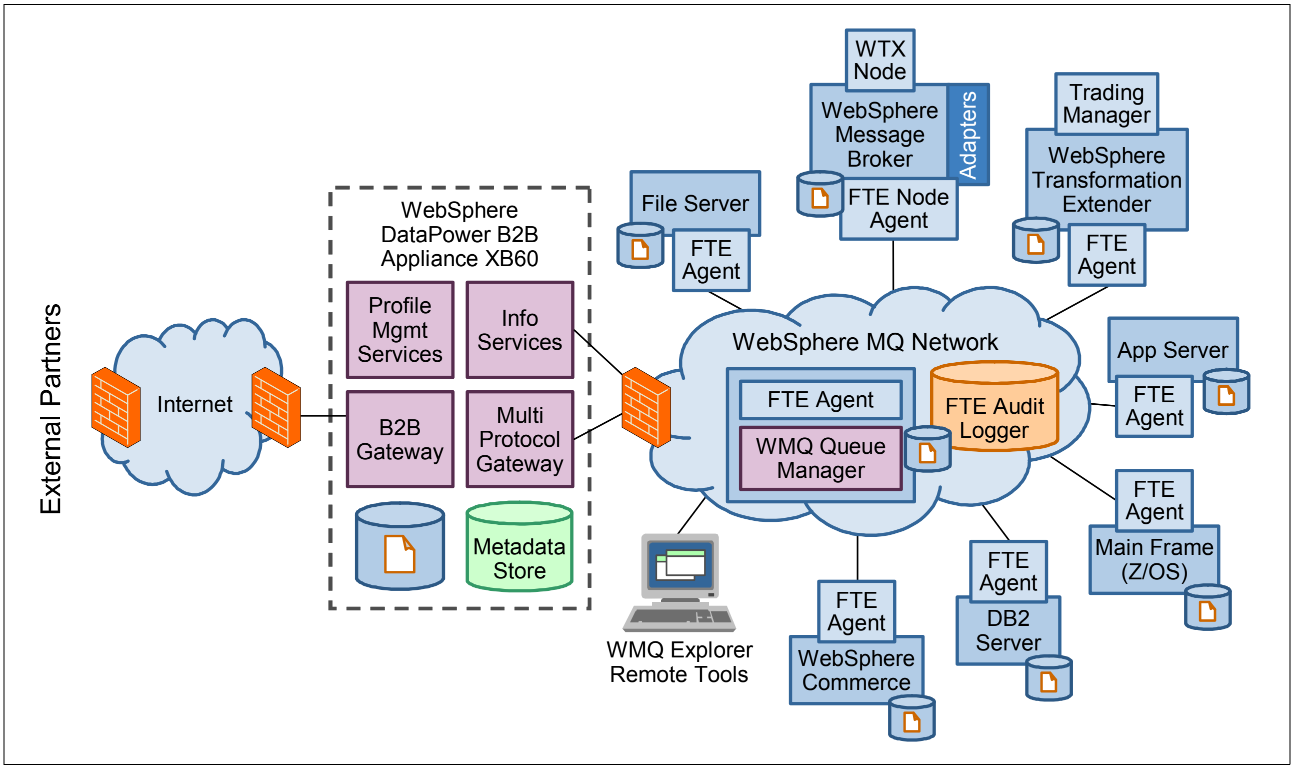


Figure 2-3 Business-to-business-enabled managed file transfer topology with IBM products

Demilitarized zone systems

The XB60 in the demilitarized zone replaces the FTP, SFTP, and HTTP servers that are typically deployed there. The XB60 builds on top of the DataPower Application Integration Appliance XI50 by adding partner profile management, B2B transaction viewing capabilities, and industry standards-based business-to-business messaging protocols to the already robust integration capabilities of the core appliance. These key capabilities are at the heart of the XB60 and are designed in such a way that the XB60 is positioned extremely well to handle simple partner connections with data passing through directly to end applications for further processing. If more complex data flows are required, the application integration capabilities of the XB60 can be used to perform data validation, transformation, rules-based enforcement, and content-based routing. For more details about the XB60 see *IBM WebSphere DataPower B2B Appliance XB60 Revealed*, SG24-7745.

Protected network systems

The systems in the protected network are identical to those listed in the application layer protocol topology. For a description of the protected network see “Protected network systems” on page 19.

3

Product overview

In this chapter we discuss IBM products that can help you build a secure file transfer solution. This chapter provides a quick technical overview of the primary products used in the following scenarios:

- 3.1, “IBM WebSphere MQ File Transfer Edition” on page 22
- 3.2, “IBM WebSphere DataPower B2B Appliance XB60” on page 28
- 3.3, “WebSphere Message Broker” on page 31

3.1 IBM WebSphere MQ File Transfer Edition

WebSphere MQ File Transfer Edition provides an enterprise-ready managed file transfer capability that is both robust and easy-to-use. WebSphere MQ File Transfer Edition exploits the proven reliability and connectivity of WebSphere MQ to transfer files across a wide range of platforms and networks. In addition to leveraging existing WebSphere MQ networks, you can easily integrate WebSphere MQ File Transfer Edition with existing file transfer systems.

The benefits that WebSphere MQ File Transfer Edition offers are:

- Auditability

  WebSphere MQ File Transfer Edition provides full logging of transfers at both the source and the destination systems. File transfer audit logs are stored in WebSphere MQ queues and optionally in a relational SQL database.

- Ease-of-use

  Using WebSphere MQ File Transfer Edition, you can initiate file transfers using a graphical user interface in WebSphere MQ Explorer, command-line commands, and scripts.

- Simplicity

  WebSphere MQ File Transfer Edition has a low resource footprint and, apart from WebSphere MQ, has no other prerequisite software.

- Security

  Access to files is controlled by file-system permissions. File transfers can be protected using SSL encryption and authentication.

- Automation

  File transfers can be set up to occur at specified times or dates or repeated at specified intervals. File transfers can also be triggered by a range of system events, such as new files or updated files.

3.1.1 Architecture overview

WebSphere MQ File Transfer Edition comprises a number of components that are all supported by one or more WebSphere MQ queue managers in the network (Figure 3-1).

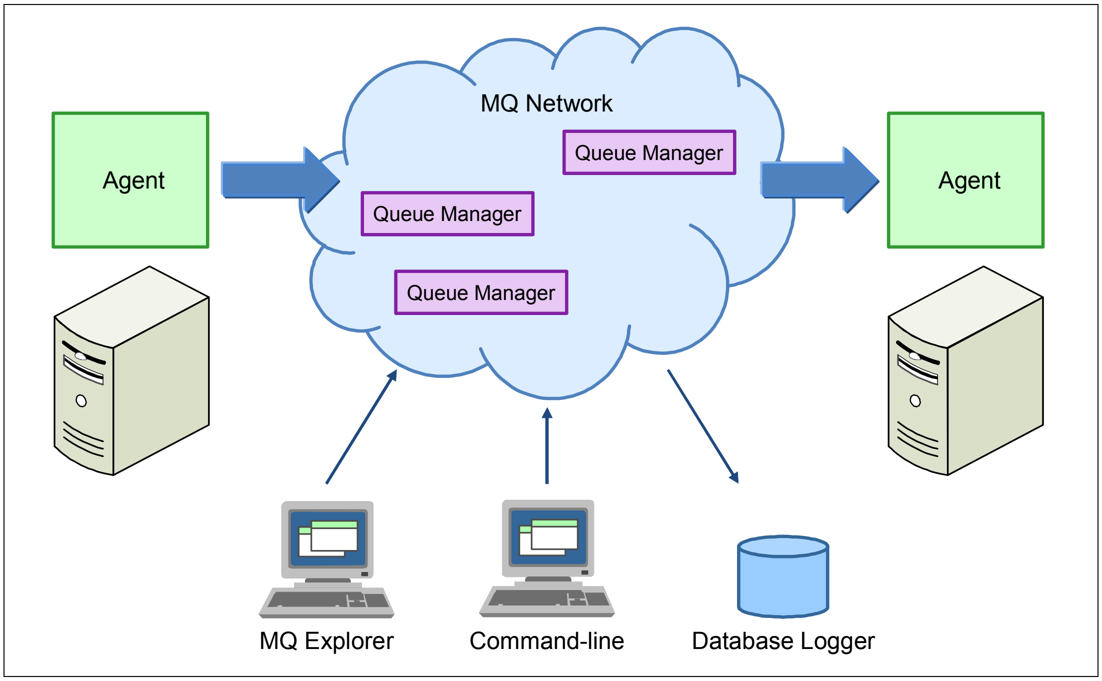


Figure 3-1 Overview of WebSphere MQ File Transfer Edition Architecture

These components are:

- *FTE agents*, which perform the fundamental file transfer function. For example, they send and receive files from the local system.
- *Configuration commands*, which are used to control FTE from a command line. Configuration commands perform tasks such as creating and deleting agents.
- *Administration commands*, which perform tasks such as creating new file transfers.
- *Graphical user interface*, which provides a point-and-click graphical interface to configure and administer FTE.
- *Database Logger*, which sends the contents of file transfer log messages to a database.

The components of WebSphere MQ File Transfer Edition use WebSphere MQ to communicate with each other, and the agents use WebSphere MQ to transport the contents of files across the network to other agents.

WebSphere MQ File Transfer Edition agents

Agents are Java™-based MQ programs that form the endpoints for file transfer operations. Essentially, agents perform the fundamental task of transferring files across the network using a WebSphere MQ network as the backbone. When requested to send a file, an agent reads the file's contents and sends it to the destination agent in the form of MQ messages. Often, those messages are carried by a WebSphere MQ channel across the network, where another agent receives them. The receiving agent re-assembles the file on the destination system.

There must be an FTE agent running on each host system that can transfer files to or from other systems. A single agent can process more than one file transfer concurrently, and concurrent transfers might be to the same or other destination agents.

Agents use the WebSphere MQ network to send file information, so every agent needs a queue manager, which is called an *agent queue manager.* An agent queue manager can host more than one agent because each agent uses its own queues, which are separate from the queues that other agents use.

There are two types of FTE agents that correspond to the IBM WebSphere MQ File Transfer Edition Server product and the IBM WebSphere MQ File Transfer Edition Client product:

- Server Edition agent

  The agent supplied with the FTE Server edition product can connect to a local queue manager using an *MQ bindings connection*. These agents can also connect to a local or remote queue manager using an *MQ client connection*.

- Client Edition agent

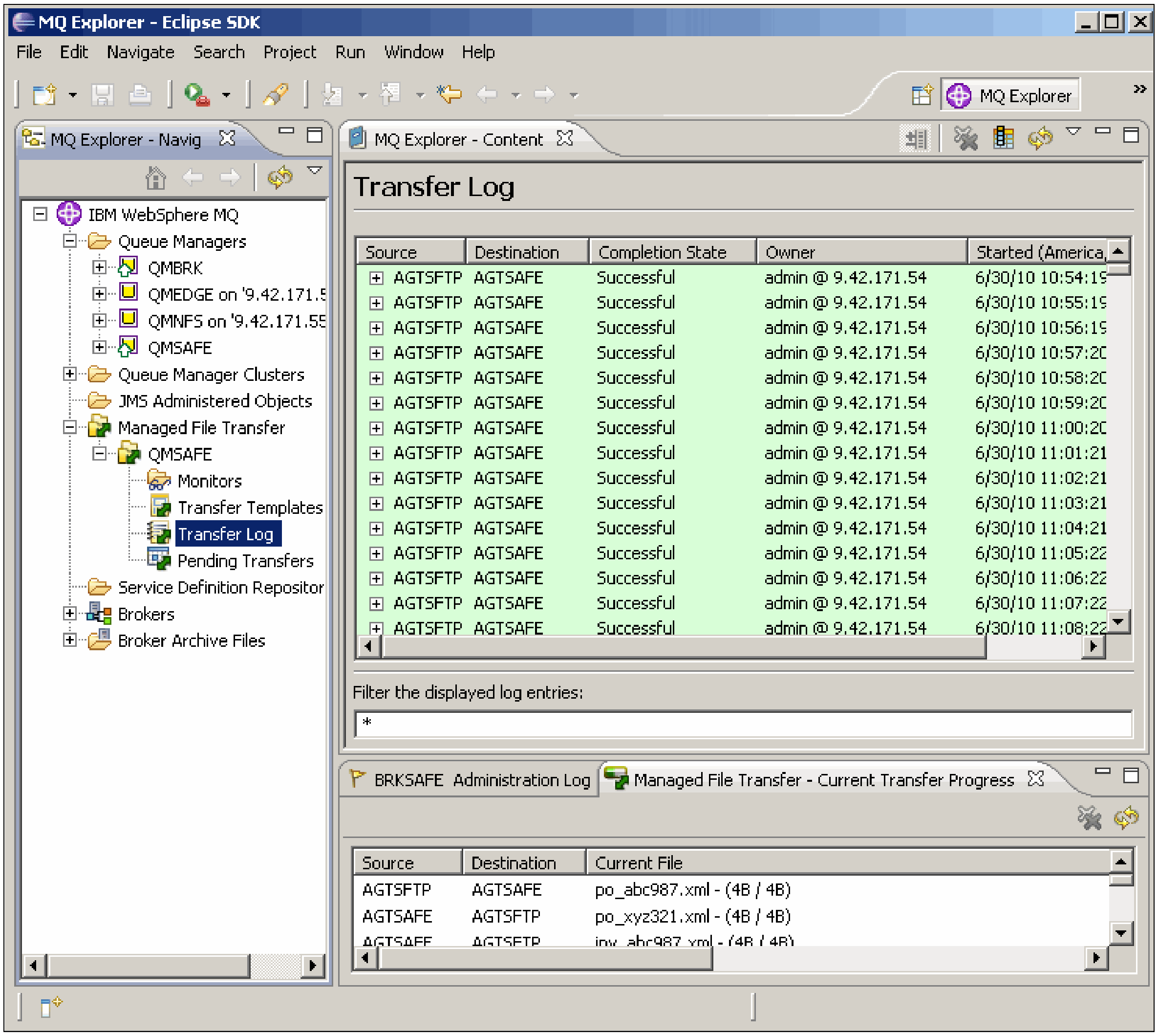
  The agent that is supplied with the FTE Client edition product uses an *MQ client connection* to connect to a queue manager. Client agents can be located on the same system or on a separate system from their agent queue manager.

A third type of agent, the *bridge agent*, is available in the Server edition. The bridge agent implements a protocol bridge that allows the WebSphere MQ File Transfer Edition network to access files stored on an FTP/SFTP file server outside the network. All agents involved in a transfer that includes the protocol bridge must be WebSphere MQ File Transfer Edition Version 7.0.1 or later. The bridge agent must be at WebSphere MQ File Transfer Edition Version 7.0.2.

Graphical user interface

You can administer WebSphere MQ File Transfer Edition with the WebSphere MQ File Transfer Edition Explorer workbench, a GUI plug-in for WebSphere MQ Explorer that is included on the IBM WebSphere MQ File Transfer Edition Remote Tools and Documentation DVD. WebSphere MQ Explorer is available for Windows® and Linux platforms, supplied with WebSphere MQ, and in stand-alone form with WebSphere MQ MS0T SupportPac.

Figure 3-2 shows the WebSphere MQ Explorer views for managing WebSphere MQ File Transfer Edition when the plug-in is installed.



Source	Destination	Completion State	Owner	Started (America,
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 10:54:19
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 10:55:19
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 10:56:19
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 10:57:20
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 10:58:20
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 10:59:20
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 11:00:20
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 11:01:21
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 11:02:21
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 11:03:21
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 11:04:21
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 11:05:22
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 11:06:22
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 11:07:22
⊞ AGTSFTP	AGTSAFE	Successful	admin @ 9.42.171.54	6/30/10 11:08:22

Source	Destination	Current File
AGTSFTP	AGTSAFE	po_abc987.xml - (4B / 4B)
AGTSAFE	AGTSFTP	po_xyz321.xml - (4B / 4B)
AGTSAFE	AGTSFTP	inv_abc987.xml - (4B / 4B)

Figure 3-2 Administering WebSphere MQ File Transfer Edition Using WebSphere MQ Explorer

Command-line tools

You can use the command-line tools to configure WebSphere MQ File Transfer Edition and to operate it (for example, submit and monitor file transfer requests).

The configuration commands are used to set up WebSphere MQ File Transfer Edition and to create and configure a new installation. The commands are:

- **`fteCreateAgent`**
- **`fteDeleteAgent`**
- **`fteChangeDefaultConfigurationOptions`**
- **`fteSetupCoordination`**
- **`fteSetupCommands`**
- **`fteCreateBridgeAgent`**

The administration commands are used to operate WebSphere MQ File Transfer Edition and support tasks that are typically performed day-to-day in an installation:

- **`fteStartAgent`**
- **`fteCreateTransfer`**
- **`fteCreateMonitor`**
- **`fteDeleteMonitor`**
- **`fteListAgents`**
- **`fteShowAgentDetails`**
- **`fteListScheduledTransfers`**
- **`fteDeleteScheduledTransfer`**
- **`fteCleanAgent`**
- **`fteStopAgent`**
- **`fteCancelTransfer`**
- **`fteSetAgentTraceLevel`**
- **`fteListMonitors`**
- **`fteAnt`**
- **`ftePingAgent`**
- **`fteStartDatabaseLogger`**
- **`fteStopDatabaseLogger`**

Queue managers

WebSphere MQ File Transfer Edition uses WebSphere MQ for communicating between its agents, the WebSphere MQ File Transfer Edition Explorer plug-in, and the command-line commands. Furthermore, WebSphere MQ File Transfer Edition uses WebSphere MQ to transmit bulk file data between agents.

To do its job, each component needs to connect to a WebSphere MQ queue manager, of which there are three roles:

- Coordination queue manager
- Agent queue managers
- Command queue managers

WebSphere MQ File Transfer Edition does not require that all three roles be physically separate queue managers, although there are good reasons to design your installation that way. A simple installation can designate a single queue manager to fill all three roles, but doing this in a production environment is not ideal from a performance and reliability point-of-view.

Production environments are best designed using separate coordination and agent queue managers.

The coordination queue manager

The coordination queue manager acts as a central collection point where information about file transfer activity is gathered. A WebSphere MQ File Transfer Edition network typically has a separate designated queue manager as the coordination queue manager. Agents publish active file transfer status information to a topic that is hosted on this queue manager. Additionally, the coordination queue manager broadcasts file transfer audit information to other components and to any interested and authorized parties that subscribed to WebSphere MQ File Transfer Edition information topics.

The coordination queue manager's primary role is to collect information about the network, and unless the coordination queue manager is also hosting WebSphere MQ File Transfer Edition agents, it does not participate in the transmission of file data (the agent queue managers perform that duty). Of course, it is possible to define a single queue manager that fills both the coordination queue manager role and the agent queue manager role, and in that case, the coordination queue manager also carries file data.

WebSphere MQ File Transfer Edition requires that the coordination queue manager be hosted using a WebSphere MQ V7 or later installation. Additionally, the coordination queue manager must be enabled for WebSphere MQ publish/subscribe.

The agent queue manager

Each agent connects to its agent queue manager and through it receives file transfer requests and publishes its own file transfer start and stop status events to the coordination queue manager.

An agent queue manager hosts the queues that are used by the agents that it supports. Each agent uses its own uniquely named set of queues so that an agent queue manager can support one or more server agents on its local system, in addition to one or more client agents on remote systems.

The command queue manager

The command-line tools and the WebSphere MQ File Transfer Edition Explorer plug-in use the command queue managers to communicate with agents.

3.1.2 Using Apache Ant

Ant is an XML-based scripting tool, which the Apache Software Foundation releases, that is widely used for building Java-based software suites. Although its original purpose was to manage building Java software, Ant is becoming popular as a general-purpose scripting tool. WebSphere MQ File Transfer Edition can integrate its file transfer functions using scripts that are run by Ant.

Ant accepts a script file that is coded in XML. Within the XML script are verbs, known as Ant *tasks*, that represent the actions that the script will perform. Ant itself provides many hundreds of tasks to address a wide range of scripting needs.

WebSphere MQ File Transfer Edition provides its own set of Ant tasks that can be used to integrate file transfer processing within an Ant script. The WebSphere MQ File Transfer Edition tasks can be combined with any of the other Ant tasks to address more complex file management needs.

WebSphere MQ File Transfer Edition provides the following Ant tasks:

- awaitoutcome
- call
- cancel
- filecopy
- filemove
- ignoreoutcome
- ping
- uuid

A number of the Ant tasks that WebSphere MQ File Transfer Edition provides use nested XML elements to further qualify the operations.

3.1.3 Using file transfer pre-processing and post-processing tasks

When you configure WebSphere MQ File Transfer Edition to send and receive files, it is possible to have WebSphere MQ File Transfer Edition run a task both before and after the transfer occurs. Pre-processing tasks are executed before the file transfer, and post-processing tasks are executed after the transfer.

Additionally, you can configure pre-processing and post-processing tasks for either the source agent or the destination agent, or both.

3.2 IBM WebSphere DataPower B2B Appliance XB60

IBM WebSphere DataPower B2B Appliance XB60 simplifies, helps secure, and accelerates your business-to-business trading partner connectivity. The XB60 is a purpose-built B2B gateway for simplified deployment and hardened security. This 1 U (1.75-inch thick) rack-mountable network device is powered by unique technology to help your business:

- Easily manage and connect to trading partners using industry standards.
- Extend integration beyond the enterprise with a securely deployed B2B gateway in the DMZ.
- Improve the performance and scalability of business-to-business interfaces.
- Govern business-to-business integration points through consolidated trading partner management.

The XB60 is based on the DataPower Application Integration XI50 appliance, which provides highly manageable, security-enhanced, and scalable SOA solutions. The XI50 includes Enterprise Service Bus (ESB) capabilities, data enablement and integration features, and the ability to improve web services management and SOA governance.

The XB60 builds on top of the DataPower Application Integration appliance by adding trading partner profile management, B2B transaction viewing capabilities, and industry standards-based business-to-business messaging protocols to the already robust integration capabilities of the core appliance. These three key capabilities are at the heart of the XB60. They are designed in such a way that the XB60 is positioned extremely well to handle simple partner connections with data passing through directly to end applications for further processing. If more complex data flows are required, the application integration capabilities of the XB60 can be used to perform data validation, transformation, rules-based enforcement, and content-based routing.

3.2.1 Features and benefits

IBM WebSphere DataPower B2B Appliance XB60 delivers secure trading partner data integration tracking, routing, and security functions in a network device, cutting operational costs and improving performance. The XB60 is a nondisruptive technology that allows organizations to extend their existing business-to-business implementations and internal integration infrastructure, thus delivering rapid return on investment and reduced total cost of ownership. The benefits are:

- Trading partner management for business-to-business governance:
  - Business-to-business protocol policy enforcement
  - Access control
  - Message filtering
  - Data security
- Application integration with stand-alone B2B gateway capabilities supporting business-to-business patterns for AS2, AS3, and web services.
- A full-featured user interface for B2B configuration and for transaction viewing that correlates documents and acknowledgments and allows you to display all associated events.
- Simplified deployment, configuration, and management, providing a quicker time to value by establishing rapid connectivity to trading partners.
- Full hardware ESB capability, including:
  - Acceleration of existing integration hubs
  - Mainframe modernization and web services
  - Any-to-any transformation
  - Integrated message-level security
  - Sophisticated multi-step message routing, filtering, and processing
  - Multiple synchronous and asynchronous transport protocols
  - Configurable quality of service
  - Detailed logging and audit trail
  - Standards-based interfaces
  - Agile, highly flexible underlying scripting/configuration support
  - XML enablement and wirespeed application integration
  - Metadata-based integration

3.2.2 B2B gateway architecture overview

Figure 3-3 depicts the components that make up the B2B gateway object in the XB60.

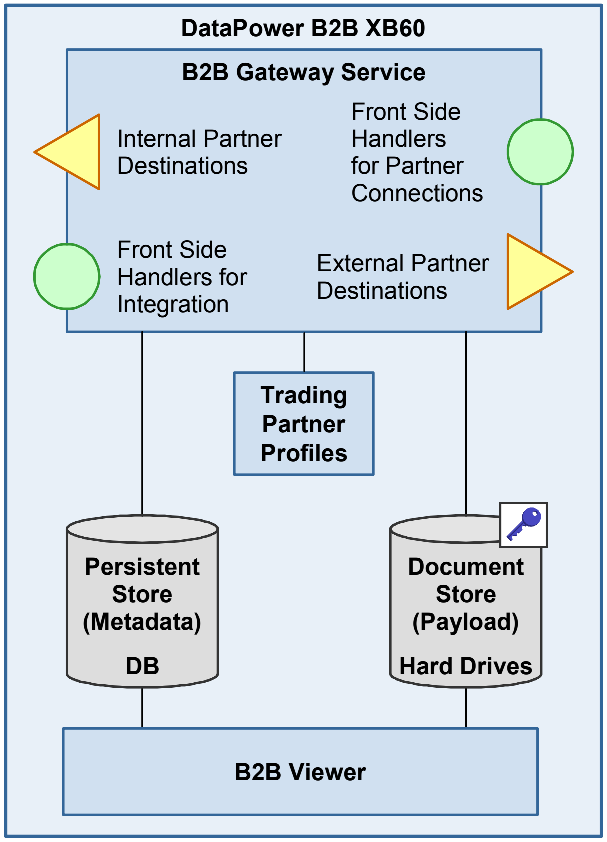

DataPower B2B XB60
B2B Gateway Service
Internal Partner Destinations
Front Side Handlers for Partner Connections
Front Side Handlers for Integration
External Partner Destinations
Trading Partner Profiles
Persistent Store (Metadata)
DB
Document Store (Payload)
Hard Drives
B2B Viewer

Figure 3-3 XB60 B2B components

The components to note are:

- The B2B Gateway service is a configuration object that is responsible for processing and routing business-to-business data.
- Partner profiles are configuration objects that are capable of supporting multiple destinations. The profiles are associated with any number of B2B Gateway services.
- The B2B Viewer is used to view all transactions that pass through a B2B Gateway service.

IBM WebSphere DataPower B2B Appliance XB60 supports the following business-to-business functionality:

- B2B Gateway service
  - AS2 and AS3 packaging/unpackaging
  - Non-repudiation of origin and receipt
  - Message Disposition Notifications (MDNs)
  - EDI, XML, and binary payload routing
  - Front-side protocol handlers
  - Trading partner profile management
  - Multiple destinations (back-side protocol handlers)
  - Certificate management (security)
  - Hard drive archive/purge policy

- B2B transaction viewer
  - Transaction viewing
  - Transaction resend capabilities
  - Acknowledgement correlation
  - Transaction event correlation
  - Role-based access
- Persistent storage
  - Encrypted with a box-specific key
  - B2B document storage
- Persistent transaction store
  - B2B metadata storage
  - B2B state management

3.3 WebSphere Message Broker

WebSphere Message Broker is a platform-independent enterprise service bus (ESB) that provides universal connectivity. It can be used to integrate disparate applications and is designed to transform various formats of data between any type of applications using a number of supported communications protocols or distribution methods. It is used where there is a need for high-performance and complex integration patterns.

3.3.1 Message flows with WebSphere Message Broker

Processing logic in WebSphere Message Broker is implemented using *message flows*. Through message flows, messages from business applications can be transformed and routed to other business applications. Message flows are created by connecting *nodes* together. A wide selection of built-in nodes is provided with WebSphere Message Broker. These nodes perform tasks that are associated with message routing, transformation, and enrichment. The base capabilities of WebSphere Message Broker are enhanced by SupportPacs that provide a wide range of additional enhancements.

Message routing

Packaged with WebSphere Message Broker is a variety of nodes through which connectivity is provided for both standards and non-standards-based applications and services. Routing can be point-to-point or based on matching the content of the message with a pattern that is specified in a node.

Aggregation is an advanced form of message routing. With aggregation, a request message is received, and multiple new request messages are generated. Each new message is routed to its destination using a request-reply interaction. WebSphere Message Broker tracks the process, collecting all the responses and recomposing them into a single output message.

Transport protocol conversion

WebSphere Message Broker provides universal connectivity between applications that use disparate transport protocols. WebSphere Message Broker enables connectivity between applications or business processes that use transport protocols, such as web services (SOAP, REST), HTTP(S), Java Message Service (JMS), WebSphere MQ, CICS®, IMS™, TCP/IP, FTP, SCA, EIS (SAP, Siebel, PeopleSoft), and user-defined transports.

WebSphere Message Broker also supports integration with WebSphere Business Adapters. For more information about available adapters, see the WebSphere Adapters page at:

http://www-01.ibm.com/software/integration/wbiadapters/

Message transformation and enrichment

One of the key capabilities of WebSphere Message Broker is the transformation and enrichment of in-flight messages. This capability enables business integration without the need for additional logic in the applications. For example, an application that generates messages in a custom format can be integrated with an application that only recognizes XML. This capability provides a powerful mechanism to unify organizations because business information can now be distributed to applications that handle completely separate message formats.

WebSphere Transformation Extender can be integrated into the WebSphere Message Broker ESB solution to extend the existing capabilities and to simplify transformation development.

3.3.2 Runtime architecture of WebSphere Message Broker

WebSphere Message Broker comprises a development environment in which message flows and message sets are designed and developed and a runtime environment on which the message flows execute.

The *broker* is a set of application processes that host and run message flows. When a message arrives at the broker from a business application, the broker processes the message before passing it on to one or more other business applications. The broker routes, transforms, and manipulates messages according to the logic that is defined in its message flow applications. Each broker uses an internal repository on the local file system to store the message flows, configuration data, and the message sets that are deployed to it.

Execution groups are processes that host message flows. The execution groups facilitate the grouping of message flows within the broker with respect to functionality, load balancing, and other qualifications that are determined to be necessary.

3.3.3 Developing message flows with the WebSphere Message Broker Toolkit

The WebSphere Message Broker Toolkit is an integrated development environment (IDE) and graphical user interface (GUI) based on the Eclipse platform. Application developers use the WebSphere Message Broker Toolkit to create message flows and the associated artifacts and to deploy them to the execution groups.

The Broker Application Development perspective is the default perspective that displays the first time that you start the WebSphere Message Broker Toolkit. Application developers work in this perspective to develop and modify message sets and message flows.

3.3.4 Deploying message flow applications

Message flow applications contain message flows and the message sets that comprise the message definitions that are used to model the messages within the message flows. Message flow applications can be deployed to the execution groups of the brokers by first adding the components of the message flow application to a *Broker Archive file* (BAR file) and then deploying the bar file to the broker's execution group. You can deploy the BAR files using the command line or with the WebSphere Message Broker Toolkit. Alternatively, WebSphere Message Broker Toolkit provides the capability to deploy the message flows directly to an execution group without first adding them to a BAR file, which results in the deployment of all of the message flow dependent resources.

3.3.5 Administration with WebSphere Message Broker Explorer

The WebSphere Message Broker Explorer is a tool for administrators and provides the capability for enhanced WebSphere Message Broker monitoring and management. WebSphere Message Broker Explorer is installed as a plug-in for WebSphere MQ Explorer. The Brokers view is added to the WebSphere MQ Explorer Navigator pane. The broker administration tasks can then be performed from this Brokers view. Using the WebSphere Message Broker Explorer, the user can administer their brokers and WebSphere MQ queue managers in the same toolkit.

In addition to the WebSphere Message Broker Explorer, there are command-line and API-based utilities that you can use to accomplish broker administration tasks. For example, the CMP API Exerciser sample that WebSphere Message Broker provides utilizes the Java administration API. You can use the CMP API Exerciser to view and manage brokers and their execution groups. You can also use it to create, modify, and delete configurable services and general broker administration tasks. The Java administration API is also available to users for scripting broker administration tasks.

Part 2

File transfer scenarios

This part contains five file transfer scenarios that illustrate the concepts discussed in Part 1, “Multi-enterprise file transfer concepts” on page 1:

- Chapter 4, “Scenario topology overview” on page 37
- Chapter 5, “Initiating file transfers outside your enterprise with HTTPS” on page 43
- Chapter 6, “Using FTP/SFTP with WebSphere MQ File Transfer Edition” on page 85
- Chapter 7, “B2B-enabled managed file transfer” on page 109
- Chapter 8, “Integrating partner transfers with internal ESB” on page 175

4

Scenario topology overview

In this book, we primarily focus on how to use WebSphere MQ File Transfer Edition and the DataPower XB60 for multi-enterprise file transfers. We also demonstrate how WebSphere MQ File Transfer Edition can work with Hypertext Transfer Protocol (HTTP) and Secure File Transfer Protocol (SFTP) to move files in a multi-enterprise environment through the various network zones that exist in many organizations.

This book includes four file transfer scenarios, with additional variations, that illustrate how you can build a file transfer solution. Each scenario addresses specific needs and environmental considerations. This chapter introduces these scenarios. It also shows the systems topology used in our lab environment, the location of the products within that topology, and basic information about the configuration. Each scenario uses a subset of the components in the topology.

4.1 An introduction to the scenarios

The scenarios in this book illustrate how to design file transfer solutions that address a variety of situations. Each scenario uses technology to suit specific needs and, in certain cases, variations of the scenario are shown. You can use these solution designs as they are, or you can use them as the basis to design your own solution. The scenarios in this book are:

- Initiating file transfers outside your enterprise with HTTPS

  This scenario illustrates how to initiate and monitor the transfer of files using an HTTP(s) client, such as a web browser or mobile device. This scenario is based on the use of a web gateway to accept file transfers from the web client. The gateway uses a RESTful API to send the file the MQ FTE backbone.

  The web gateway is provided by the FO02: WebSphere MQ File Transfer Edition - Web Gateway. The SupportPac includes a sample application to illustrate the interface for the web client. Both the web gateway and the sample application run on WebSphere Application Server.

  The scenario addresses the flow of files from an external partner into the enterprise.

- Using FTP/SFTP with WebSphere MQ File Transfer Edition

  This scenario illustrates how WebSphere MQ File Transfer Edition can integrate with external partners that use File Transfer Protocol (FTP) or SFTP. A special kind of WebSphere MQ File Transfer Edition agent called a bridge agent is used to facilitate this integration. This agent sits in the protected network.

  There are two topologies shown for this scenario:

  - In the first topology, an SFTP server sits in the DMZ. Files transferred into and out of the enterprise are temporarily stored in the DMZ. The bridge agent moves files between the protected network and the temporary file store in the DMZ.
  - In the second topology, files are transferred between the external partner and the protected network without stopping in the DMZ.

  Both topologies support inbound and outbound file transfer flow.

- B2B-enabled managed file transfer

  This scenario illustrates how files can be exchanged between enterprises using business-to-business messaging protocols such as EDIINT AS1, AS2, and AS3 and partner profile management at the edge of the network.

  The IBM WebSphere DataPower B2B Appliance XB60 is used in the DMZ to provide the business-to-business enablement. It provides the ability to handle business-to-business protocols, as well as the profile management for partner relationships. An MQ FTE backbone inside the protected network is used to move files securely between external partners and back-end systems.

  This scenario addresses both inbound and outbound file transfers.

- Integrating partner transfers with internal ESB

  This scenario illustrates how file transfers with external partners that do not use business-to-business protocols can be achieved, while at the same time taking advantage of the business-to-business partner profile capabilities of the XB60.

  In addition, it illustrates how you can use an enterprise service bus (ESB) in the protected network to affect the file transfer. In this scenario, WebSphere Message Broker provides the ESB capabilities, allowing you to route data to the appropriate back-end application, to transform the transfer request to the protocol required for the target application, or to perform more advanced mediation using the wide range of capabilities of WebSphere Message Broker.

This scenario addresses both inbound and outbound file transfers.

4.2 Scenario architecture

The product components used throughout this book reside on multiple systems that are located in specific network locations within and external to the enterprise. There are three network zones involved in each file transfer solution:

- The Internet, where external partners reside
- The protected network that contains the applications and data belonging to the enterprise
- The demilitarized zone (DMZ) that acts as a security buffer between the two

The zones are designed to simulate the common zones found in organizations. Figure 4-1 depicts a view of the complete scenario architecture.

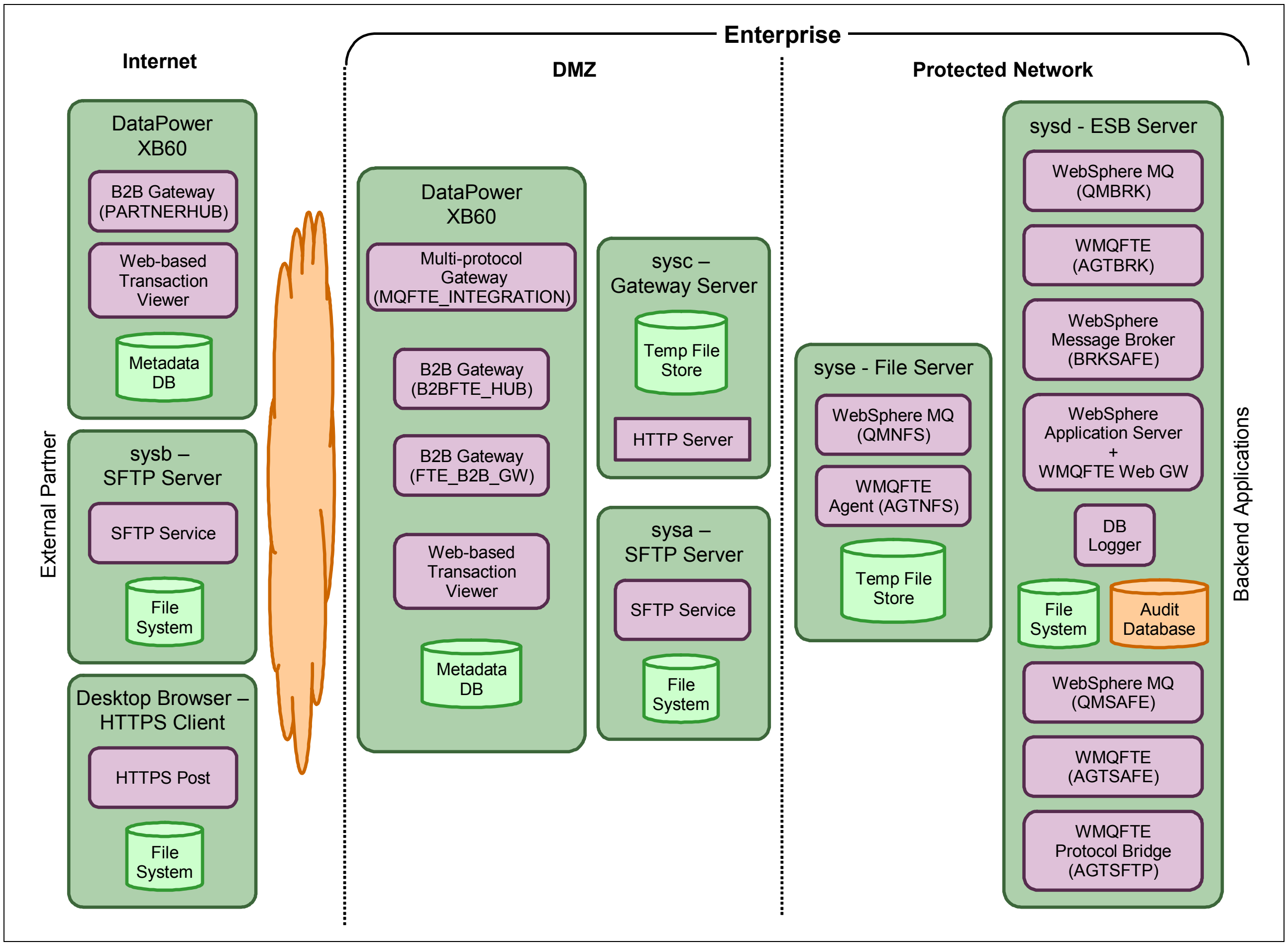


Figure 4-1 Multi-Enterprise File Transfer Edition with WebSphere Connectivity Systems Architecture

The systems and components shown in Figure 4-1 are discussed here at a high level, along with basic implementation information where appropriate. As specific alterations are made to the environment for each file transfer scenario, the specific modification or implementation details made to that system are shown in the corresponding file transfer scenario chapter. Many of the middleware components reside on separate systems but use the same installation and configuration steps. The following sections provide basic information about the topology configuration.

4.2.1 The internet

In this book, when we refer to the internet, we are referring to anything outside of the enterprise's span of control. We have placed an SFTP server, a business-to-business server, and an HTTP client on the internet to represent our file transfer partners. The SFTP server consists of a SFTP service and file system. The business-to-business server in our case is the IBM WebSphere DataPower B2B Appliance XB60 with a B2B Gateway service configured.

External partner

An external partner is an independent entity that trades data with an entity in the protected network. The trading can take the form of business-to-consumer, business-to-business, or business-to-government transfers. For the purposes of this book, the external partner represents any location where you move a file to that you have no control over. This can be inside your enterprise firewalls in a separate department or outside the firewalls at a separate organization. The external partner is used in all scenarios.

SFTP server

The SFTP server uses an open Secure Shell (SSH) daemon. For certain operating systems, this is part of the base operating system. For our purposes, we used a Linux server and started the SFTP service built into the operating system.

IBM WebSphere DataPower B2B Appliance XB60

To simulate a business-to-business trading partner, an XB60 was configured with a B2B gateway, web-based transaction viewer, and metadata database. The B2B gateway is designed to handle the packaging and unpackaging of business-to-business protocols such as AS2. Additionally, the B2B Gateway service allows for trading partner management. Running on the XB60 is a service for the web-based transaction viewer that allows interested parties, both internal and external, to view the results of their transfer.

HTTP client

Simply stated, the HTTP client is a web browser. This can be any web browser available today. The web browser is on a desktop where it can read or write to the file system using HTTP verbs.

4.2.2 The demilitarized zone

The demilitarized zone (DMZ) hosts technologies designed specifically for placement in a DMZ. In our scenarios, we use the WebSphere DataPower B2B Appliance XB60 for trading partner management, packing and unpacking business-to-business protocols, viewing external transactions, and security. Our scenarios also make use of a SFTP server and the IBM HTTP Server in the DMZ.

IBM WebSphere DataPower B2B Appliance XB60

Providing trading partner management, the XB60 uses a multiprotocol gateway service, a B2B Gateway service, and a web-based transaction viewer. The B2B Gateway service processes and routes business-to-business data from and to the external partner and provides partner profile management. The multiprotocol gateway routes and processes any data type. These two gateways are used in conjunction to provide end-to-end file transfer capabilities in a variety of scenarios.

In the scenario in Chapter 7, “B2B-enabled managed file transfer” on page 109, a B2B Gateway service named B2BFTE_HUB interfaces with the multiprotocol gateway, MQFTE_INTEGRATION, to move business-to-business data across the MQ FTE backbone.

In Chapter 8, “Integrating partner transfers with internal ESB” on page 175, a B2B Gateway service named B2B_FTE_GW interfaces with the multiprotocol gateway, MQFTE_INTEGRATION, to move non-business-to-business data across the MQ FTE backbone, where it is further mediated with WebSphere Message Broker acting as an enterprise service bus.

IBM HTTP Server

The IBM HTTP Server is a web server that terminates connections from the Internet in the DMZ. It is hardened and designed for placement in the DMZ. Any web server or proxy server can be used.

In Chapter 5, “Initiating file transfers outside your enterprise with HTTPS” on page 43, the IBM HTTP Server receives the HTTP requests for file transfer. The WebSphere Web server plug-in is installed on the IBM HTTP server and has been configured to send files over an SSL-enabled channel to WebSphere Application Server.

Configuration details for the IBM HTTP Server and Web server plug-in can be found in Appendix B, “Preparing the WebSphere Application Server and IBM HTTP Server environment” on page 297.

SFTP server

The SFTP server is implemented by simply using the Open SSH service featured on the operating system. Any secure FTP or SSH service can be used.

4.2.3 The protected network

Hosted in the protected network are:

- WebSphere Application Server
- WebSphere MQ
- WebSphere Message Broker
- WebSphere MQ File Transfer Edition
- DB2®

WebSphere MQ File Transfer Edition

WebSphere MQ File Transfer Edition is used in every scenario to provide an MQ FTE backbone for files to traverse on their way to their destination. The agents reside exclusively in the protected network and are used to push and pull files from the DMZ, and in certain cases, directly from the Internet. For information about how the WebSphere MQ File Transfer Edition agents are configured, see Appendix A, “Configuration of WebSphere MQ File Transfer Edition” on page 269.

WebSphere MQ

WebSphere MQ is a messaging product available on a wide range of platforms. In these scenarios, WebSphere MQ provides the messaging network for WebSphere MQ File Transfer Edition. For information about how we configured the WebSphere MQ infrastructure required for the MQ FTE backbone, see Appendix A, “Configuration of WebSphere MQ File Transfer Edition” on page 269.

WebSphere Application Server

The WebSphere Application Server is a Java Platform, Enterprise Edition (JEE) application server developed on Open Standard Specifications. In Chapter 5, “Initiating file transfers outside your enterprise with HTTPS” on page 43, the WebSphere Application Server is used to run the application that translates HTTP verbs to WebSphere MQ File Transfer Edition commands. For configuration details for the application server, see Appendix B, “Preparing the WebSphere Application Server and IBM HTTP Server environment” on page 297.

WebSphere Message Broker

WebSphere Message Broker is a powerful information broker that allows business data, in the form of messages, to flow between disparate applications and across multiple hardware and software platforms. Rules can be applied to the data that is flowing through the message broker to route, store, retrieve, and transform the information.

WebSphere Message Broker provides a choice of transports that enable secure business to be conducted at any time by providing powerful integration, message, and data transformations in a single place. WebSphere Message Broker is built on WebSphere MQ. Therefore, it supports MQ transport. However, it also supports many other transports, such as HTTP/HTTPS, SOAP, file, TCP/IP, EIS, and others, that do not use MQ stack.

WebSphere Message Broker is used in multi-enterprise file transfer to transform and route files. For more information about its use and configuration, see Chapter 8, “Integrating partner transfers with internal ESB” on page 175.

DB2 Enterprise Server Edition

DB2 Enterprise Server Edition is a relational database developed by IBM. DB2 Enterprise Server Edition is utilized in a multi-enterprise file transfer environment as the WebSphere MQ File Transfer Edition database, depicted as the audit database in Figure 4-1 on page 39. The database receives audit messages from the WebSphere MQ File Transfer Edition database logger. Any supported relational database can be used for the WebSphere MQ File Transfer Edition database. We assume in the scenarios that your relational database of choice is installed and configured.

For more information about which relational databases can be used with WebSphere MQ File Transfer Edition view the System Requirements for WebSphere MQ File Transfer Edition:

http://www-01.ibm.com/software/integration/wmq/filetransfer/requirements/

5

Initiating file transfers outside your enterprise with HTTPS

In this chapter we describe one of the simplest and most easily accessible interfaces for transferring files between an organization and its internal and external users: a web client.

This scenario allows a web client application to transfer files by connecting, via an Hypertext Transfer Protocol (HTTP) proxy server, to a web gateway application integrated with the organization's WebSphere MQ File Transfer Edition Backbone.

5.1 Scenario overview

It is a common requirement in most organizations to provide HTTP(S) web gateway connections to their enterprise. In the case of file transfers using WebSphere MQ File Transfer Edition, such a web gateway provides the capability to extend an existing WebSphere MQ File Transfer Edition network to support clients that use the HTTP(S) protocol.

The purpose of this scenario is to provide an example of how one might use a Web Gateway API to provide web gateway services. The specific API used is included in the FO02: WebSphere MQ File Transfer Edition - Web Gateway SupportPac, found at:

http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24026419&loc=en_US&cs=utf-8&lang=en

Support: The FO02 SupportPac is classified as a Category 2 SupportPac. These types of SupportPacs are provided in good faith and as is. There is no warranty or further service implied or committed to, and any supplied sample code is not supported by IBM product service channels.

5.1.1 Appropriate use

A web gateway is useful in situations in which you have files to be transferred on a system where you cannot use a WebSphere MQ File Transfer Edition agent but can use an HTTP(S) client. For example, you can use a web gateway for the following tasks:

- Sending files to a WebSphere MQ File Transfer Edition agent using a web page that is running in a web browser
- Monitoring the status of transfers using a web page that is running in a web browser
- Sending files from a portable device that is not capable of running the WebSphere MQ File Transfer Edition infrastructure but that has HTTP capabilities
- Sending files from an operating system on which an WebSphere MQ File Transfer Edition agent is not supported

It is important when using such a web gateway that appropriate role-based security measures are in place—that backend resources are appropriately isolated from direct web access.

5.1.2 Business value

Web gateway access to an organization’s WebSphere MQ File Transfer Edition infrastructure provides low-cost, simple, platform-diverse, and easily customizable access for purposes of file transfer. This approach serves both an organization’s geographically dispersed internal users and its external partners. It provides access to the organizations’s WebSphere MQ File Transfer Edition managed file transfer infrastructure via commonly available web clients. Additionally, the development of such web clients relies on development skills that are commonly available.

5.2 Scenario details

The FO02 Web Gateway SupportPac upon which this scenario is based provides a RESTful API to WebSphere MQ File Transfer Edition. Its purpose is to provide an API that can be used to build user web solutions that involve transferring data or viewing the outcome of file transfers.

This API provides the following capabilities:

- Initiating a file transfer from an HTTP client (for example, a web browser) to a WebSphere MQ File Transfer Edition agent including:
  - Setting typical transfer options (for example, transfer priority, text/binary transfer type)
  - Receiving a unique ID relating to the transfer, which can be used to track the transfer through the FTE system, including querying the status of the transfer
- Retrieving audit information about a file transfer operation. Reported information includes:
  - The success of individual files in the transfer
  - The amount of data transferred and the time that the transfer occurred

The API provides a set of HTTP verbs and supported WebSphere MQ File Transfer Edition parameters that allow an application developer to implement the above functionality. For more information about the SupportPac and the Web Gateway API, see:

`ftp://public.dhe.ibm.com/software/integration/support/supportpacs/individual/fo02.pdf`

5.2.1 Solution components

This section describes the components associated with each product in this solution (Figure 5-1). Certain components require specific configuration for the solution to work. We discuss the configuration steps required when necessary.

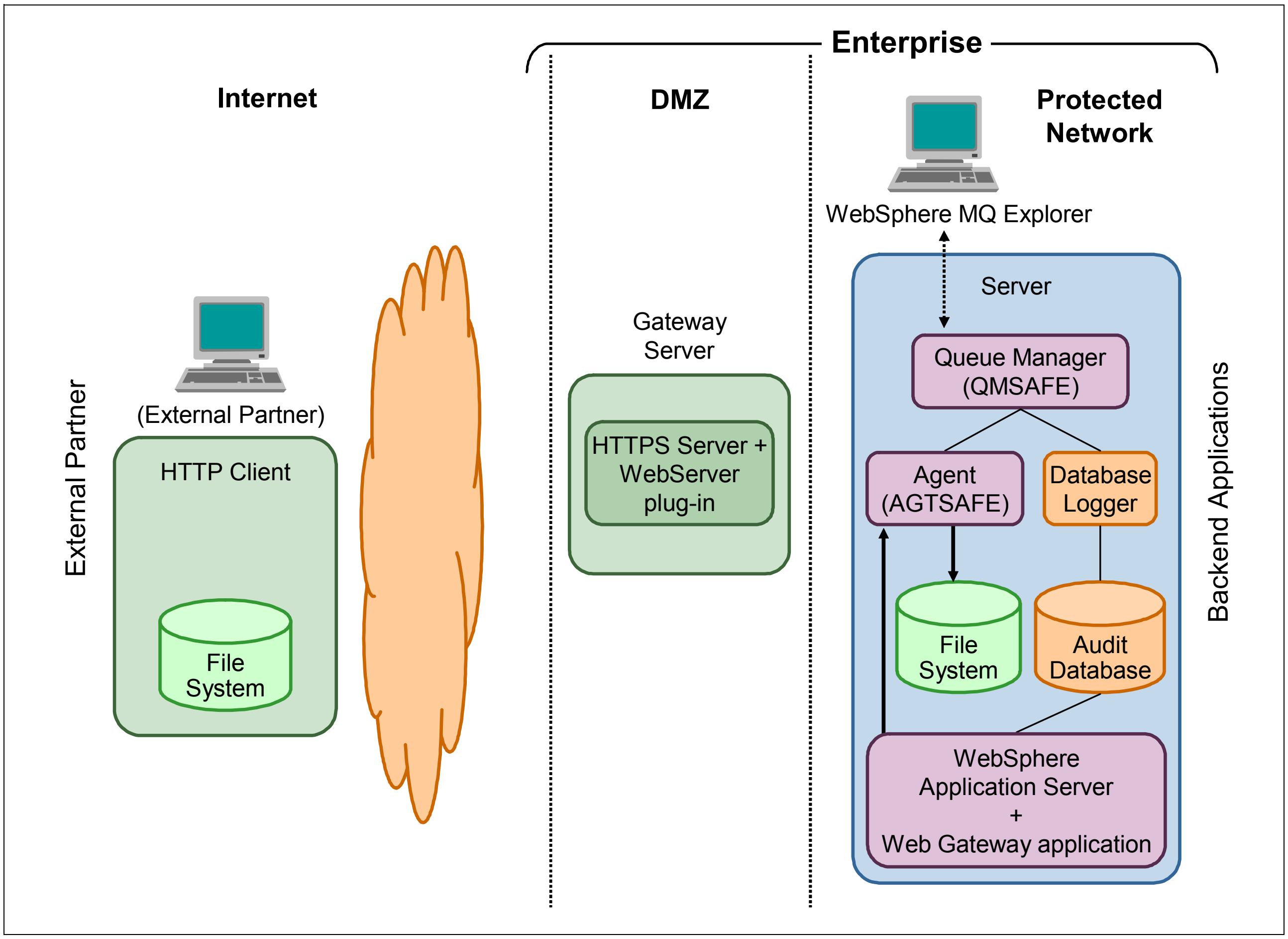


Figure 5-1 Scenario components

WebSphere MQ queue manager (QMSAFE)

Queue manager QMSAFE hosts the WebSphere MQ File Transfer Edition agent AGTSAFE on the backend server. This server is the endpoint for the file transfer. QMSAFE is the coordination queue manager for the file transfer network. QMSAFE is also the command and agent queue manager for AGTSAFE.

WebSphere MQ File Transfer Edition Server Agent (AGTSAFE)

AGTSAFE is a WebSphere MQ File Transfer Edition agent in the protected network. The agent connects to the local queue manager, QMSAFE, using bindings mode. This bindings connection is indicative of a server agent. This agent reads and writes files on the local file system. AFTSAFE can connect with other WebSphere MQ File Transfer Edition agents to send or receive files from between file systems.

WebSphere MQ File Transfer Edition Database Logger

The database logger is a Java-based MQ application that creates a subscription, either administratively or an through API, to the coordination queue manager. The subscription messages are then off-loaded to the tables created in a relational database to store audit information.

WebSphere MQ Explorer

The WebSphere MQ Explorer is used to view and administer the WebSphere MQ queue managers and queue manager objects such as queues, topics, and channels. WebSphere MQ Explorer is built on an Eclipse integrated development environment. The Eclipse-based platform allows plug-ins to be added to the base platform.

WebSphere MQ File Transfer Edition Explorer

The WebSphere MQ File Transfer Edition Explorer is a plug-in to the WebSphere MQ Explorer. It is used to schedule file transfer requests and view the status of current requests. The tool includes a Transfer Log view that subscribes to the coordination queue manager to obtain audit information. The audit information is displayed in the Transfer Log view for every transfer that occurs in the given topology.

WebSphere Application Server

WebSphere Application Server Network Deployment is used in this scenario. A standalone application server is created for simplicity. The application server hosts the WebSphere MQ File Transfer Edition Web Gateway and sample application supplied in the FO02 SupportPac. The application server makes a Java Database Connectivity (JDBC) connection to the audit database to gather information regarding the transfer.

IBM HTTP Server and Web server plug-in

The IBM HTTP Server receives the HTTP requests for file transfer. The WebSphere Web server plug-in is installed on the IBM HTTP server and has been configured to forward requests over an Secure Socket Layer (SSL)-enabled channel to a Web gateway application running on the stand-alone application server.

DB2

The DB2 relational database is used to store audit information published by the coordination queue manger. The audit messages are loaded into a user-created database and tables by the database logger.

SupportPac FO02: WebSphere MQ File Transfer Edition Web Gateway

The FO02 SupportPac enables files to be transferred to an agent through HTTP commands. It also allows users to check the status of a transfer by passing requests to the audit database via HTTP commands. The Web Gateway and sample application contained in the SupportPac run on a JEE application server.

Figure 5-2 shows a high-level view of the components supported by the FO02 Web Gateway SupportPac API. Web applications can be written in a variety of languages such as Perl, Ruby, Python, or Java. The application runs on the client's machine and communicates with WebSphere MQ File Transfer Edition using a WebSphere MQ File Transfer Edition Web Gateway application running in an application server. A client can initiate the transfer of a file to an agent, and then receive status information that has been recorded in a database.

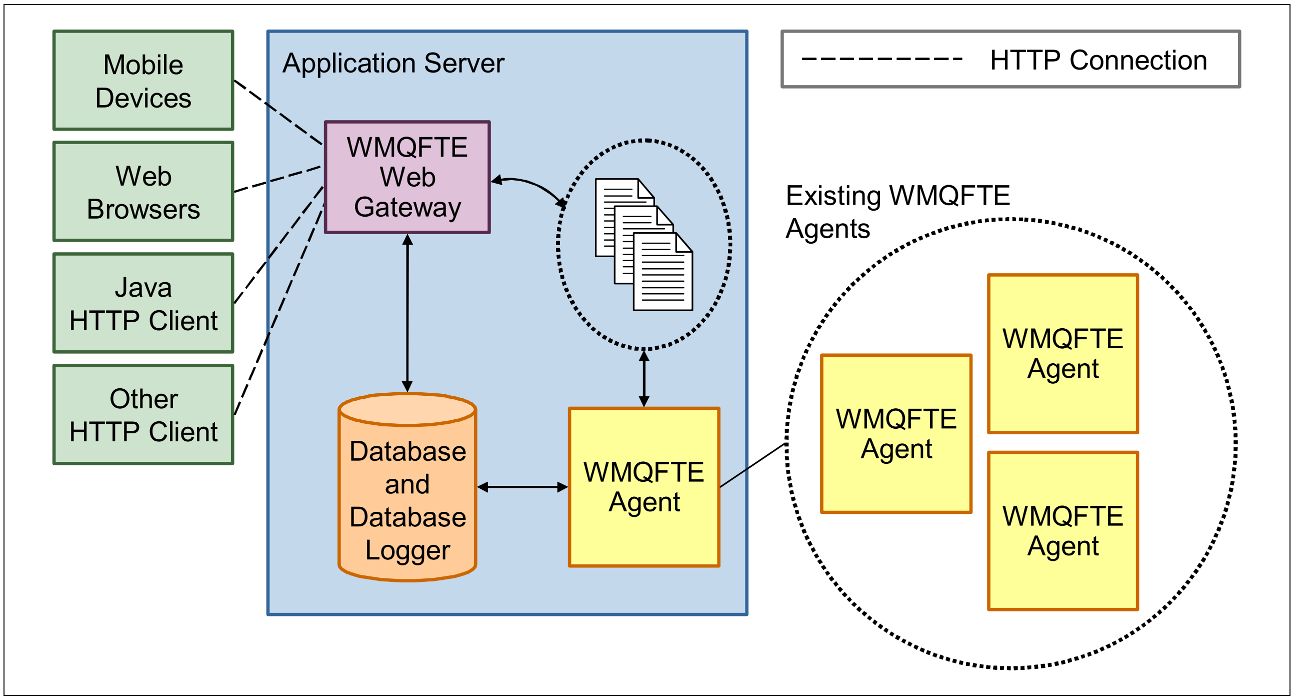


Figure 5-2 Overview of FO02 SupportPac Web Gateway API Architecture

Included with the FO02 Web Gateway SupportPac is a sample Web Gateway application that was built using the FO02 Web Gateway API. It is this sample application that is used in this scenario.

Figure 5-3 shows a detailed flow of the use of the FO02 sample Web Gateway application and how it serves to link a web client user to an WebSphere MQ File Transfer Edition backend infrastructure.

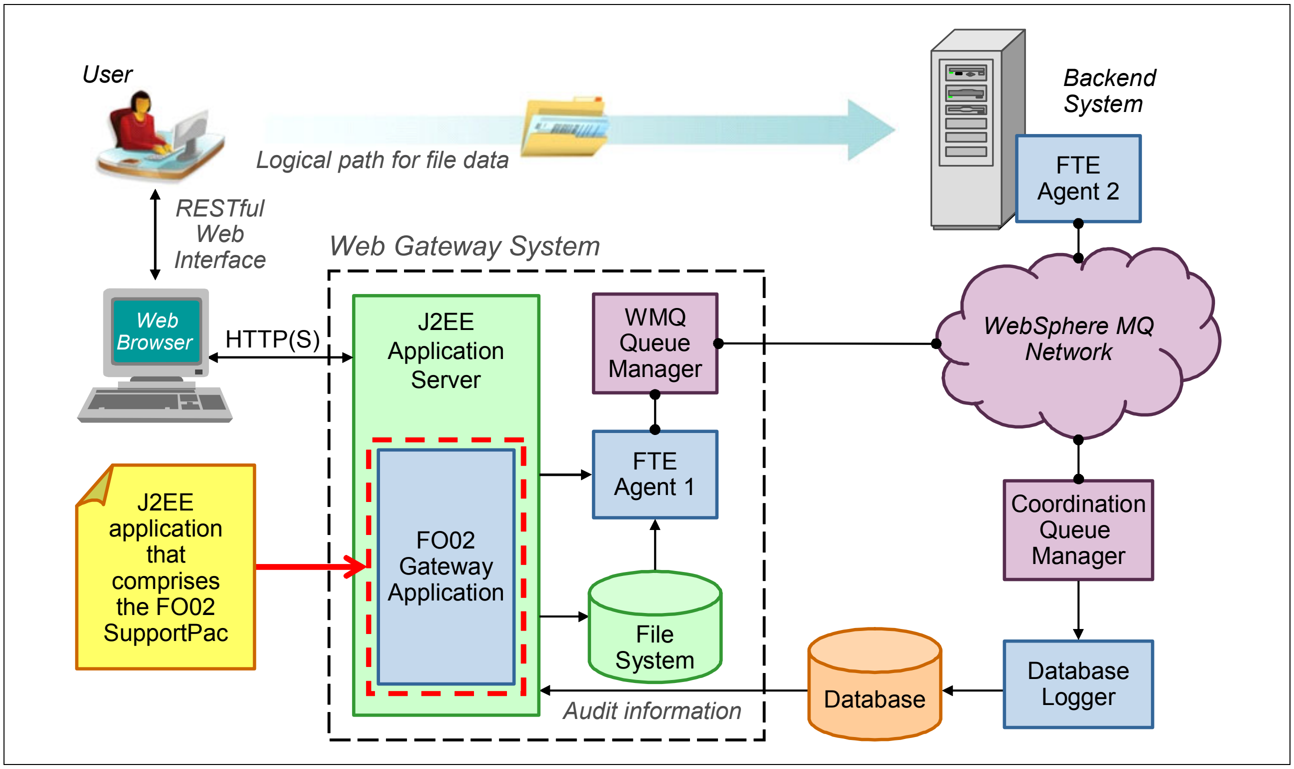


Figure 5-3 Topology of FO02 SupportPac sample Web Gateway Application

The goal of this flow is to get a file to a backend system without giving the web application access to the backend.

The user specifies the following values:

- The files to be transferred
- The destination agent (agent 1)

This information is used by the gateway application to communicate with the WebSphere MQ File Transfer Edition backbone and transfer the file to a local file system. The status of the transfer, captured by the coordination queue manager, is stored in a database by the WebSphere MQ File Transfer Edition Database Logger. The gateway application reports this transfer status back to the user. The backend system can then use agent 2 to pull the transferred file to the backend system.

The sample application does a great job of showing how the Web Gateway can be used. However, the sample application is not designed for production environments. Organizations can create their own production application using the HTTP verbs given through the Web Gateway. For more information about the verbs and parameters for HTTP headers, consult the `fo02.pdf` file in the FO02 SupportPac.

Web Gateway support for inbound and outbound flows: The 2010-04-29 version of the FTEO02 Web Gateway SupportPac supports the following HTTP verbs:

- POST: Upload files and create a new transfer.
- GET: Retrieve the status of a previous transfer.

Using these verbs, we can build a secure solution for inbound file transfer, but not outbound file transfer. Check the most current version of the Web Gateway for additional functionality.

The following sections provide examples of how HTTP requests can be formed.

Example: Sending a file using an HTTP request

You can send a single text file to a destination file system by submitting a request through the Web Gateway.

The example in this section shows an HTTP request to transfer a text file to the destination file path <*destination-root-path*>/temp using the myfile.txt destination file, on the destination agent ACCOUNTS. The transfer uses an MD5 checksum to check the integrity of the transferred file. The server hosting the Web Gateway is example.com.

Example 5-1 shows the HTTP request. In this example, the file transmission is a literal HTTP message. The contents of the HTTP message are put into a file named myfile.txt for the purpose of sending the transfer. This type of HTTP request is done programmatically instead of by a user. Ideally, the content is something that the code populates (possibly through a database query or user entry) and then sends to a file for transfer.

Example 5-1 Sample HTTP request

```
POST HTTP/1.1 /fte/file/agent/ACCOUNTS/temp
Host: example.com
Content-Type: multi-part/form-data; boundary=Aa6b74
x-fte-checksum: MD5
--Aa6b74
Content-Disposition: form-data; name="files"; filename="myfile.txt"
Content-Type: text/plain
Account No, Balance
123456, 100.00
234567, 1022.00
345678, 2801.00
456789, 16.75
--Aa6b74
```

Example 5-2 shows the response to the HTTP request. The value of x-fte-id in the response represents the transfer ID.

Example 5-2 Sample HTTP response

```
HTTP/1.1 200 OK
Server: WAS/6.0
Content-Length: 0
x-fte-id: 4d63c28ae6e72eb9c51cd812736acd4362ef5
<transfers>
<submission id=Ë4d63c28ae6e72eb9c51cd812736acd4362ef5Ë>
</submission>
</transfers>
```

Example 5-3 shows the contents of the `myfile.txt` file.

Example 5-3 myfile.txt contents

```
Account No, Balance
123456, 100.00
234567, 1022.00
345678, 2801.00
456789, 16.75
```

Example: Viewing the status of file transfers using an HTTP request

You can view the status of your transfers by submitting a request through the Web Gateway. The Web Gateway returns an XML describing the current status of the specified transfer.

A successful request returns an HTTP status code of 200 and an XML payload that describes the current status of the transfer. You can use this XML to view details of the transfer, including the status of the transfer, the transfer ID, source and destination agent details, and information about the transfer's source and destination files. You can write a web application to parse the content of the XML response and display it in an appropriate format to a web user.

The following steps describe how to submit a request. In this example, the server hosting the Web Gateway is `example.com` and the HTTP request is submitted using a web browser that identifies itself as Mozilla. Example 5-4 shows the HTTP request.

Example 5-4 HTTP request

```
GET HTTP/1.1 /fte/transfer/transfer-ID
Host: example.com
User-Agent: mozilla
```

Example 5-5 shows the HTTP response returned by the Web Gateway.

Example 5-5 HTTP response

```
HTTP/1.1 200 OK
Server: WAS/6.0
Content-Length: 1664
Content-type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<transfers>
<transfer start-time="2010-04-01T13:10:04.209+01:00" status="Complete"
id="414d51205245444841542e434f4f5244ed60b44b03310020">
```

```
<source>
<agent qmgr="REDHAT.SOURCE.QM" name="REDHAT.SOURCE.AGENT" />
<metadata>
<key value="REDHAT.SOURCE.AGENT" name="com.ibm.wmqfte.SourceAgent" />
<key value="REDHAT.DEST.AGENT" name="com.ibm.wmqfte.DestinationAgent" />
<key value="192.168.243.133" name="com.ibm.wmqfte.OriginatingHost" />
<key value="fteuser" name="com.ibm.wmqfte.MqmdUser" />
<key value="414d51205245444841542e434f4f5244ed60b44b03310020"
name="com.ibm.wmqfte.TransferId" />
<key value="fteuser" name="com.ibm.wmqfte.OriginatingUser" />
</metadata>
</source>
<destination>
<agent qmgr="REDHAT.SOURCE.QM" name="REDHAT.SOURCE.AGENT" />
<metadata>
<key value="REDHAT.SOURCE.AGENT" name="com.ibm.wmqfte.SourceAgent" />
<key value="REDHAT.DEST.AGENT" name="com.ibm.wmqfte.DestinationAgent" />
<key value="fteuser" name="com.ibm.wmqfte.MqmdUser" />
<key value="192.168.243.133" name="com.ibm.wmqfte.OriginatingHost" />
<key value="fteuser" name="com.ibm.wmqfte.OriginatingUser" />
<key value="414d51205245444841542e434f4f5244ed60b44b03310020"
name="com.ibm.wmqfte.TransferId" />
</metadata>
</destination>
<stats retry-count="0" file-warnings="0" file-failures="0"
bytes-transferred="67" />
<transfer-set>
<file result-code="0" mode="text">
<source-file name="/opt/IBM/WMQFTE/install.properties">
<attribute-values last-modified="2010-03-17T16:55:17.000Z"
file-size="67" disposition="leave" checksum-method="none" />
</source-file>
<destination-file name="/tmp/install.properties">
<attribute-values last-modified="2010-04-01T13:10:04.000+01:00"
file-size="67" exists-action="error" checksum-method="none" />
</destination-file>
</file>
</transfer-set>
</transfer>
</transfers>
```

An invalid request returns an HTTP error code and a WebSphere MQ File Transfer Edition error message.

5.2.2 Scenario flow for files inbound to the protected network

Figure 5-4 depicts the inbound data flow for this scenario. It shows how the FTEO02 SupportPac sample application was implemented to allow an inbound file transfer.

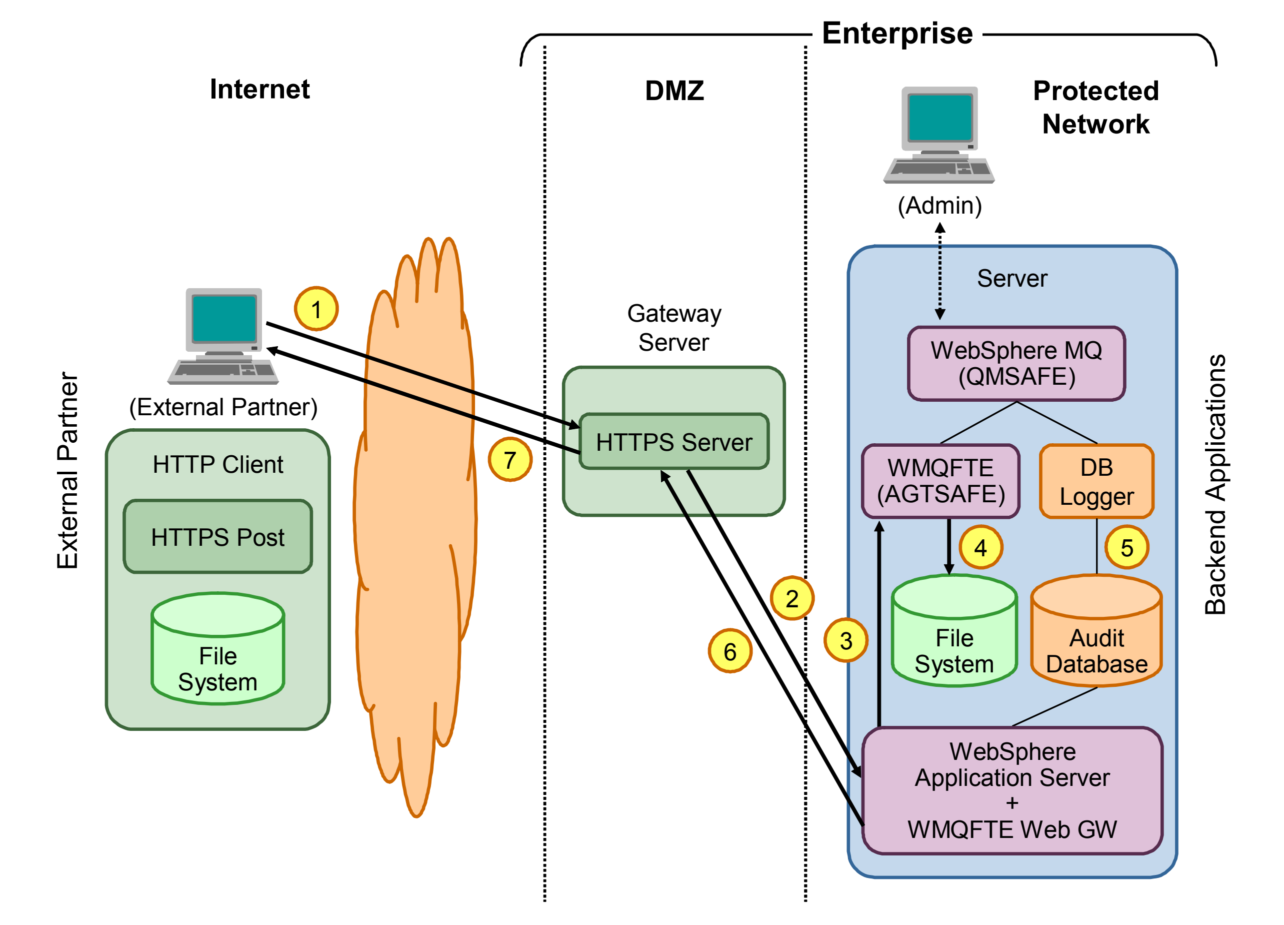


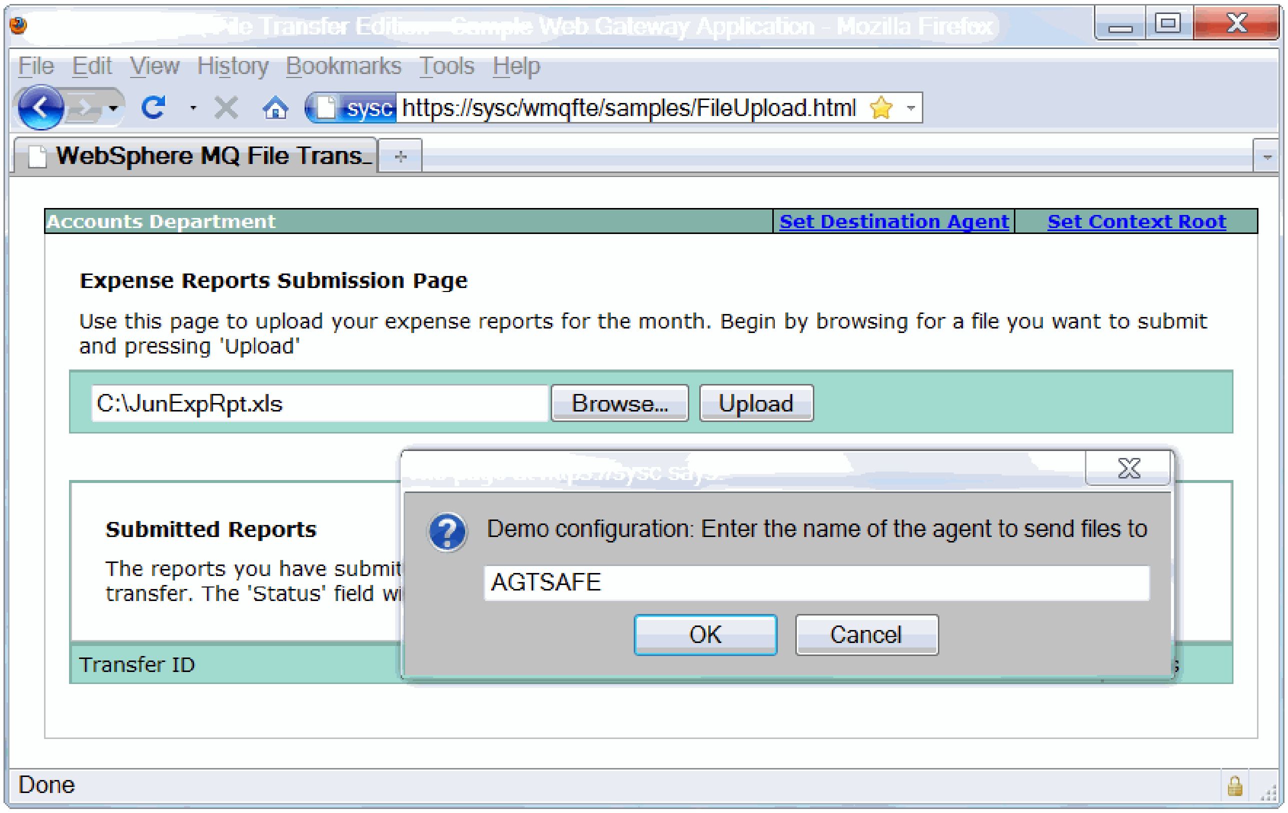
Figure 5-4 Scenario flow for inbound files transfers

The following steps are performed in this scenario:

1. Using the sample application provided in the WebSphere MQ File Transfer Edition SupportPac FO02 Web Gateway, the external partner enters the parameters required to initiate the file transfer:

   – Path to the local file to be transferred
   – Destination agent, AGTSAFE

   > **Additional parameters:** Though not implemented in the sample application, the API allows for additional WebSphere MQ File Transfer Edition parameters to be used with the Web Gateway.


File Edit View History Bookmarks Tools Help
sysc https://sysc/wmqfte/samples/FileUpload.html
WebSphere MQ File Trans...
Accounts Department
Set Destination Agent
Set Context Root
Browse...
Upload
Transfer ID
OK
Cancel
Done

Expense Reports Submission Page

Use this page to upload your expense reports for the month. Begin by browsing for a file you want to submit and pressing 'Upload'

C:\JunExpRpt.xls

Submitted Reports

The reports you have submit... transfer. The 'Status' field wi...

Demo configuration: Enter the name of the agent to send files to

AGTSAFE

Figure 5-5 File to be transferred and destination agent specified

   The completed file transfer request is sent to the HTTPS server over an encrypted connection.

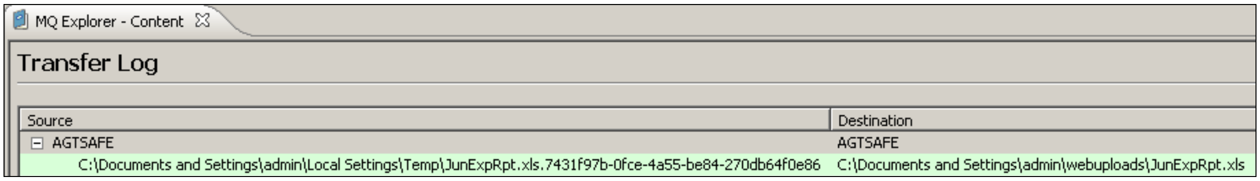
2. The HTTP Server forwards the file and associated parameters over the encrypted connection to the Web Gateway. The Web Gateway writes the file to a temporary store on the machine's file system.

3. The Web Gateway sends an XML command to the destination agent specified by the external user. The message instructs the agent to move the transferred file from the temporary store on the Web Gateway's machine to the default location given by the sample application:

   – For Linux or UNIX, the default location is `*transferRoot*/webuploads/`, where *transferRoot* is the home directory of the user ID running the destination agent process. For example, `/home/agtsafe/webuploads/`, where AGTSAFE is the agent specified by the external user, agtsafe is the user ID running the agent process, and `/home/agtsafe/` is the home directory for the user ID.

   – For Windows operating systems, the default location is `C:\Documents and Settings\*admin*\webuploads\`, where *admin* is the user ID running the agent process, and `C:\Documents and Settings\*admin*\` is the home directory for the user ID.

4. The agent executes its instruction (from the Web Gateway) to move the transferred file from the temporary store to the default location specified by the sample application. Using WebSphere MQ Explorer, this action is displayed in the WebSphere MQ File Transfer Edition Transfer Log (Figure 5-6).

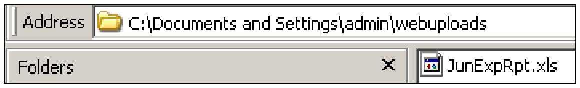


Transfer Log

Source	Destination
AGTSAFE	AGTSAFE
C:\Documents and Settings\admin\Local Settings\Temp\JunExpRpt.xls.7431f97b-0fce-4a55-be84-270db64f0e86	C:\Documents and Settings\admin\webuploads\JunExpRpt.xls

Figure 5-6 Temporary file (Source) transferred to its final destination

Figure 5-7 shows the transferred file at its final destination, as displayed using Windows Explorer.



Address: C:\Documents and Settings\admin\webuploads

JunExpRpt.xls

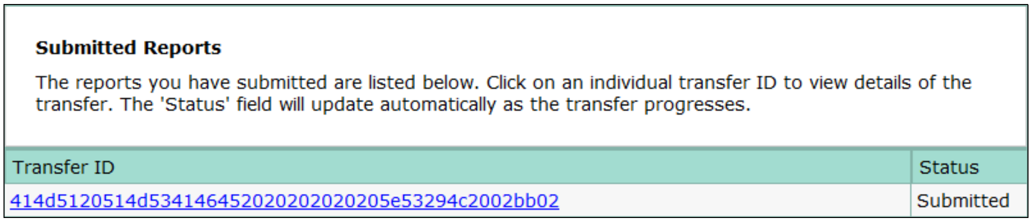
Figure 5-7 Transferred file at its final destination

5. The Database Logger writes file transfer status information to the audit database. This information includes:

   – Success or failure of the transfer of files
   – Amount of data transferred and the time that the transfer occurred
   – Destination agent and destination directory
   – The user ID associated with the transfer initiation
   – Any metadata associated with the transfer

6. The Web Gateway returns encrypted transfer status information to the HTTP server.

7. The HTTP server sends the status information to the external partner's web interface (Figure 5-8).



Submitted Reports

The reports you have submitted are listed below. Click on an individual transfer ID to view details of the transfer. The 'Status' field will update automatically as the transfer progresses.

Transfer ID	Status
414d5120514d534146452020202020205e53294c2002bb02	Submitted

Figure 5-8 File transfer ID and status of transfer

Clicking the Transfer ID link displays very detailed status information in XML format. The intent of the FO02 Web Gateway SupportPac sample application is merely to demonstrate that the API provides access to file transfer status information. It is up to the application developer, using this SupportPac API, to selectively display this status information in a user-friendly manner. The sample application, without modification, should not be used for external file transfer or in a production environment.

5.2.3 Protocols

In contrast to the architecture shown in Figure 5-3 on page 49, which shows a connection directly to a backend WebSphere MQ File Transfer Edition Agent, we adopted several measures to apply security to the file transfer process.

HTTPS versus HTTP protocol

Though the sample application supports an HTTP connection, all traffic over a public network is unencrypted. We therefore chose to implement this scenario using an HTTPS connection. HTTPS provides a secure channel over an insecure network (for example, the internet) using either a SSL or a Transport Layer Security (TLS) end-to-end connection. This provides a barrier to eavesdropping and man-in-the-middle attacks.

HTTPS proxy

The inclusion of an HTTPS proxy server in the DMZ prevents the user application from directly connecting to the backend system in the protected network.

5.2.4 Security

If you are writing an application using the FO02 Web Gateway SupportPac API, consider using the additional security measures discussed in this section.

Data integrity

The MD5 checksum should be implemented between the external partner source and the Web Gateway. This serves to detect any tampering or transmission errors occurring during the file transfer.

Role-based security

WebSphere MQ File Transfer Edition defines a number of roles that are specific to the Web Gateway. When deploying the Web Gateway to an application server, these roles can be mapped to users and groups that exist in the application server security.

WebSphere MQ File Transfer Edition, at the time of publication, defines the roles listed in Table 5-1.

Table 5-1 WebSphere MQ File Transfer Edition roles

Role	User capabilities
wmqfte-admin	► Can upload files ► Can view transfers started by other users
wmqfte-transfer	► Can upload files
wmqfte-status	► Can view transfers started by other users

For example, if the application server defines the administrator, monitor, and employee groups, then the roles can be assigned to the groups as listed Table 5-2.

Table 5-2 Application server groups and WebSphere MQ File Transfer Edition role assignments

Group	Role
administrator	wmqfte-admin
monitor	wmqfte-status
employee	wmqfte-transfer

If the application server defines the monitor group and the wmqfte-status role, which is used to restrict access to the content and resources under the `/fte/transfer/` URI address space, then the users who are in monitor group will be authorized to access the `/fte/transfer/` namespace, but others will not.

Agent access at the web GUI

If you choose to develop something similar to the sample application provided in the SupportPac, then care should be taken with the design. Any application that will use the Web Gateway should be designed to implement user-based security that associates the user ID from the logon with the appropriate application server group in order to control what WebSphere MQ File Transfer Edition functionality can be executed.

Additionally, access to agents should be controlled through the developed code using the RESTful API. The application should limit a user's ability to initiate a transfer with just any agent. The application should limit the agent names that a user can specify for the file transfer, or limit access to agents.

Agent security in the backend

Securing the agents in the protected network is of primary importance in protecting the MQ FTE backbone and its component agents. This topic is discussed in detail at the WebSphere MQ File Transfer Edition Information Center:

http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/index.jsp?topic=/com.ibm.wmqfte.admin.doc/security_main.htm

A detailed discussion of this topic is beyond the scope of this book, but a few key points are:

- Any Web Gateway application ID given access to an agent should be a member of MQ ALT_USER but *not* a member of MQM (MQ administrator group). On Windows, the user should *not* be a member of the administrators group. By placing a user in the administrators group on Windows, this automatically places the user in the mqm group.
- Make use of user groups to manage which agents and agent resources can be accessed.
- Turn on user authority checking for all agents, and then restrict user application ID access to only those agent functions required.
- Use WebSphere MQ File Transfer Edition sandboxes to restrict which areas of the file system an agent can access. You may apply restrictions to either the agent or to the user that requests the transfer. For more information about sandboxing, see:

  http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/topic/com.ibm.wmqfte.admin.doc/sandboxes.htm

- Use the commandPath property for the agent to limit which commands an agent can execute. For more information about this property, see:

  http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/topic/com.ibm.wmqfte.admin.doc/command_path.htm

- Run the agent process under its own unique user ID. This user ID should not have root or administrative authority and should not be a member of the mqm group.

5.3 Configuring the solution components

To create the Web Gateway scenario using the HTTP(S) protocol to connect external partners to the WebSphere MQ File Transfer Edition backbone, we had to install and configure an application server, create a database, configure the database logger, and install and configure a web server.

5.3.1 Prerequisites

The following software is used in this scenario:

- WebSphere MQ Version 7.0.1 or later
- WebSphere MQ File Transfer Edition Version 7.0.2
- FO02: WebSphere MQ File Transfer Edition - Web Gateway SupportPac
- WebSphere Application Server Version 7.0.0.7
- WebSphere IBM HTTP Server Version 7.0.0.7
- DB2 Version 9.5
- Web browser

> **Web browser support:** The Web Gateway was tested with Mozilla Firefox Version 3.0.1.9 and Google Chrome 4.1. Do not use Internet Explorer, as the Web Gateway has not been customized to work with that web browser.

5.3.2 Configuration prerequisites

Configure the following software before configuring the solution:

- A WebSphere Application Server environment with a running application server. See Appendix B, “Preparing the WebSphere Application Server and IBM HTTP Server environment” on page 297.
- An IBM HTTP Server with the WebSphere Application Server Web server plug-in installed. An SSL connection is established between the Web server and the application server. See Appendix B, “Preparing the WebSphere Application Server and IBM HTTP Server environment” on page 297.
- WebSphere MQ queue manager QMSAFE. See “Creating the queue managers” on page 271.
- WebSphere MQ File Transfer Edition agent AGTSAFE. See “Creating the WebSphere MQ File Transfer Edition agents” on page 281.
- Ports opened in an external fire wall: 443: Used for HTTPS to HTTP Server.

- Ports opened in an internal fire wall:
  - 14014: QMSAFE listening port
  - 9444: HTTPS port for application server
- There are no file size limitations for this scenario. You are only limited by the disk space available on the servers receiving the transfers.

5.3.3 WebSphere MQ File Transfer Edition Database Logger

The database logger sends audit information from the coordination queue manager to a relational database for archival and future reference. The Web Gateway accesses the audit database through a JDBC driver. Developers can then use their choice of API to retrieve information from the database to allow the user to see information regarding his transfer.

The configuration of the database logger is discussed in "Installing the database logger" on page 287.

After the database logger is configured and running, obtain the following information regarding the setup of the database logger for use with installing the WebSphere MQ File Transfer Edition Web Gateway:

- Database name (all platforms) or database location name (z/OS only)
- Database port number
- Location of database server (host name)
- Driver type

5.3.4 FO02 SupportPac

The FO02: WebSphere MQ File Transfer Edition - Web Gateway SupportPac enables files to be transferred to an agent and enables users to check the transfer progress using HTTP.

Web Gateway software requirements

The Web Gateway consists of several components:

- A Java Platform, Enterprise Edition 5-compliant application server

  This application server hosts the Web Gateway application. HTTP requests from clients are directed to the application server, which passes the contents of the requests to the application.

- The Web Gateway application

  This application is available in the WebSphere MQ File Transfer Edition Web Gateway application - SupportPac FO02. The Web Gateway application handles both file uploads and transfer status requests. The SupportPac is available from the following URL:

  http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24026419&loc=en_US&cs=utf-8&lang=en

- A WebSphere MQ File Transfer Edition agent

  An agent is required on the same machine as the Web Gateway. This agent receives the file transfer request message. The request message refers to the files in the temporary store. This agent then transfers the file to the specified destination agent. The source disposition behavior is to delete the file after a successful transmission.

  The system requirements and versions of WebSphere MQ File Transfer Edition are available at the following URL:

  http://www-01.ibm.com/software/integration/emq/filetransfer/requirements/

- A WebSphere MQ File Transfer Edition database logger

  The database logger provides status information about the transfers. The Web Gateway application must be able to query the database that contains the audit information for the file transfers. The database is populated by the database logger. The database does not have to be located on the same machine as any of the other components.

  The database logger is available with WebSphere MQ File Transfer Edition Version 7.0.1 or later.

  The database logger requires that the WebSphere MQ queue manager to which it connects be WebSphere MQ 7.0.0.1, 7.0.0.2, or 7.0.1.

  A relational database must be used with the database logger. The supported combinations of databases and operating systems are listed in the WebSphere MQ File Transfer Edition requirements URL.

SupportPac FO02 contents

The FO02: WebSphere MQ File Transfer Edition - Web Gateway SupportPac contains:

- `fo02.ear`

  `fo02.ear` is an enterprise archive file that contains the application code for the Web Gateway.

- `fo02.samples.war`

  `fo02.samples.war` is a web archive file that contains a sample application provided to allow users to see how the Web Gateway can be used and to test the Web Gateway.

  Usage of the sample application is shown in 5.2.2, “Scenario flow for files inbound to the protected network” on page 53.

- `fo02.pdf`

  This PDF contains documentation explaining how the Web Gateway can be installed and used. It also shows HTTP examples and common response codes.

Configuring and deploying the Web Gateway

The following steps demonstrate how to install the `fo02.ear` Web Gateway.

Define the database to the application server

Before deploying the Web Gateway application, configure the application server to access the database containing audit information loaded by the database logger. The configuration is created using the administrative console for the application server.

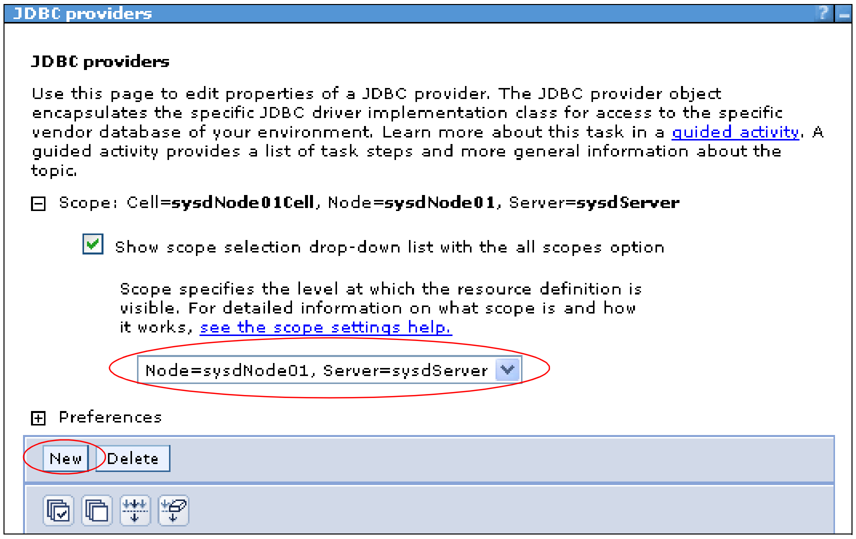
> **Setting up the database logger:** For information about setting up the database logger and creating the database, see “Configuring the database logger” on page 287.

1. Define a JDBC Provider:

   a. Select **Resources** → **JDBC** → **JDBC Providers** from the administrative console for the application server.

   b. Select the appropriate scope for your environment.

   c. Create a new JDBC provider using the console wizard by clicking **New** (Figure 5-9).



JDBC providers

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a guided activity. A guided activity provides a list of task steps and more general information about the topic.

⊟ Scope: Cell=**sysdNode01Cell**, Node=**sysdNode01**, Server=**sysdServer**

☑ Show scope selection drop-down list with the all scopes option

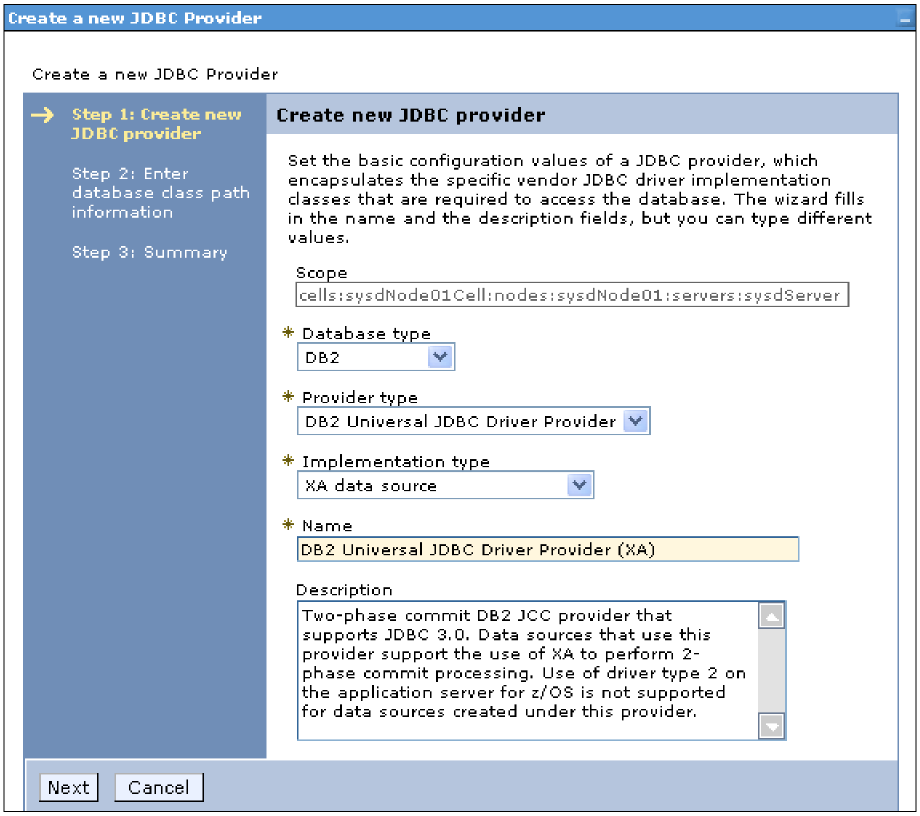
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

Node=sysdNode01, Server=sysdServer

⊞ Preferences

Figure 5-9 JDBC Provider scope and wizard

d. In Step 1 of the wizard, select the database type, provider type, and implementation type as required for the database. Click **Next**.



Create a new JDBC Provider

Create a new JDBC Provider

→ Step 1: Create new JDBC provider
Step 2: Enter database class path information
Step 3: Summary

Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope
cells:sysdNode01Cell:nodes:sysdNode01:servers:sysdServer

* Database type
DB2

* Provider type
DB2 Universal JDBC Driver Provider

* Implementation type
XA data source

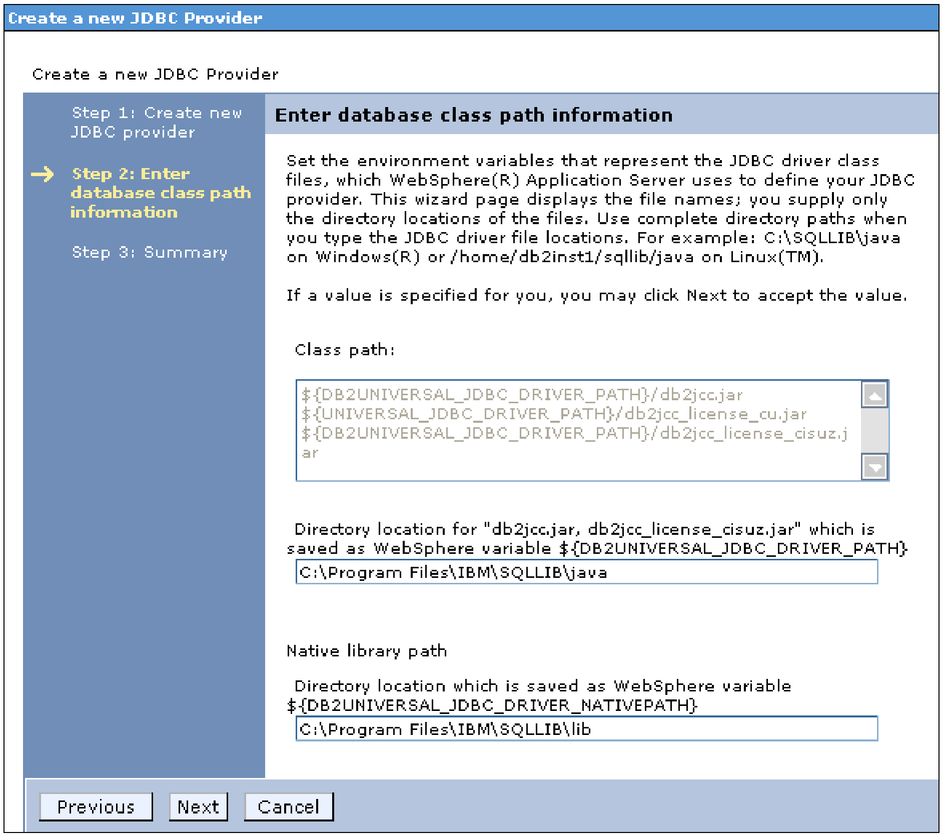
* Name
DB2 Universal JDBC Driver Provider (XA)

Description
Two-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on the application server for z/OS is not supported for data sources created under this provider.

Figure 5-10 JDBC Provider WIzard - Step 1

This application currently does not require two-phase commit, so an XA data source implementation type is not required. Connection Pool Data Source can be used as the implementation type if desired.

e. In Step 2 of the wizard, ensure that the directory location of the required database jar files is set correctly (Figure 5-11). Click **Next**.



Create a new JDBC Provider

Create a new JDBC Provider

Step 1: Create new JDBC provider

→ Step 2: Enter database class path information

Step 3: Summary

Enter database class path information

Set the environment variables that represent the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(TM).

If a value is specified for you, you may click Next to accept the value.

Class path:

```
${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar
${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar
${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.j
ar
```

Directory location for "db2jcc.jar, db2jcc_license_cisuz.jar" which is saved as WebSphere variable ${DB2UNIVERSAL_JDBC_DRIVER_PATH}

C:\Program Files\IBM\SQLLIB\java

Native library path

Directory location which is saved as WebSphere variable ${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}

C:\Program Files\IBM\SQLLIB\lib

Figure 5-11 JDBC Provider Wizard - Step 2

A local database or database connector is required on the machine where the application resides. This is required to obtain the JAR files and native files necessary for the JDBC Provider.

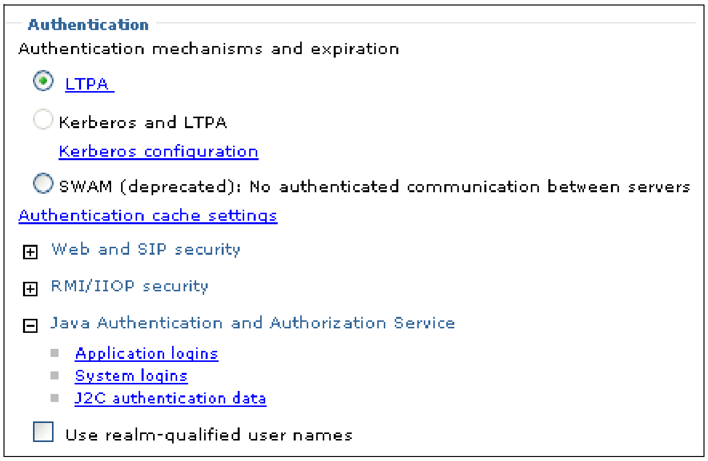
Native library path files are not needed unless a local, type 2, connection is being made to the database.

f. Click **Finish** on the summary page to create the JDBC provider.

2. Create a component-managed authentication alias:

   a. Select **Security** → **Global Security** from the left navigational menu.

   b. Expand **Java Authentication & Authorization Service** and select **J2C Authentication Data** (Figure 5-12).



Authentication

Authentication mechanisms and expiration

● LTPA

○ Kerberos and LTPA

Kerberos configuration

○ SWAM (deprecated): No authenticated communication between servers

Authentication cache settings

⊞ Web and SIP security

⊞ RMI/IIOP security

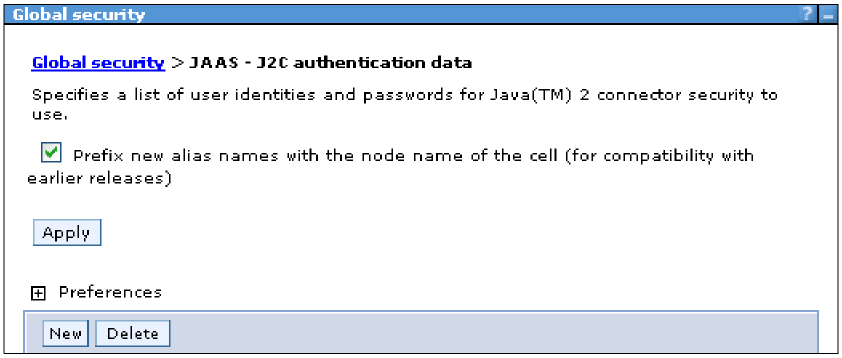
⊟ Java Authentication and Authorization Service

- Application logins
- System logins
- J2C authentication data

□ Use realm-qualified user names

Figure 5-12 Where to begin creating J2C Authentication data

   c. In the JAAS - J2C authentication data window, select **New** to create new authentication data (Figure 5-13).



Global security

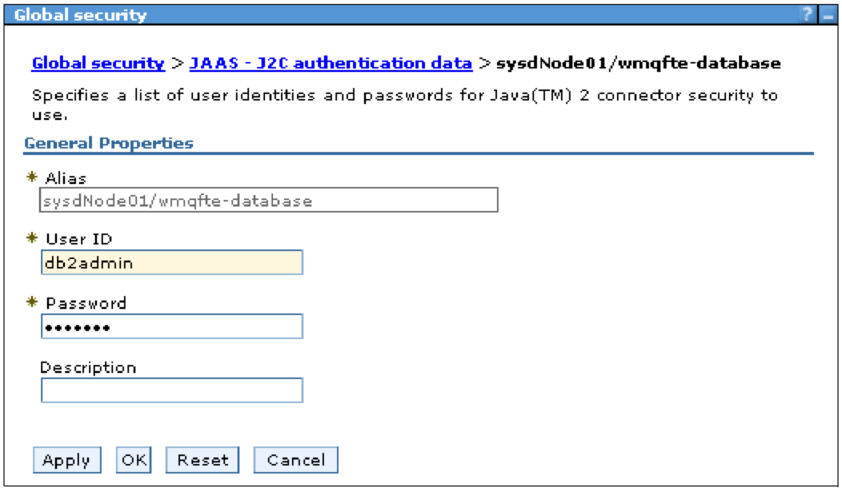
Global security > JAAS - J2C authentication data

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

☑ Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

Figure 5-13 JAAS-J2C authentication data panel

d. In the next panel, enter an alias name to be used to identify this alias definition. Enter a user ID with authority to access the database and its password (Figure 5-14).



Global security

Global security > JAAS - J2C authentication data > sysdNode01/wmqfte-database

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias
sysdNode01/wmqfte-database

* User ID
db2admin

* Password
•••••••

Description

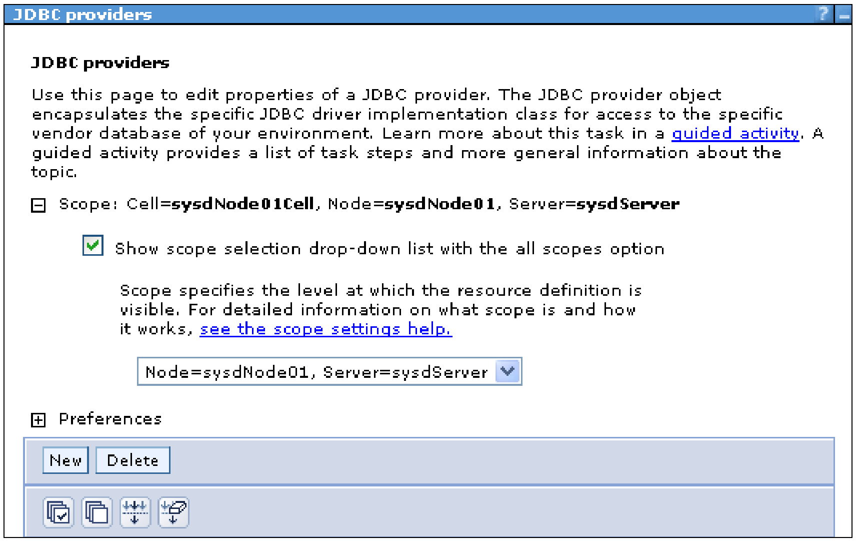
Figure 5-14 JAAS-J2C Authentication Data

Click **OK** and save the changes to the configuration.

3. Define a data source:

   a. Select **Resources** → **JDBC** → **DataSources** from the administrative console navigation.

   b. Select the scope for the data source from the drop-down menu, and then click **New** (Figure 5-15).



JDBC providers

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a guided activity. A guided activity provides a list of task steps and more general information about the topic.

⊟ Scope: Cell=**sysdNode01Cell**, Node=**sysdNode01**, Server=**sysdServer**

☑ Show scope selection drop-down list with the all scopes option

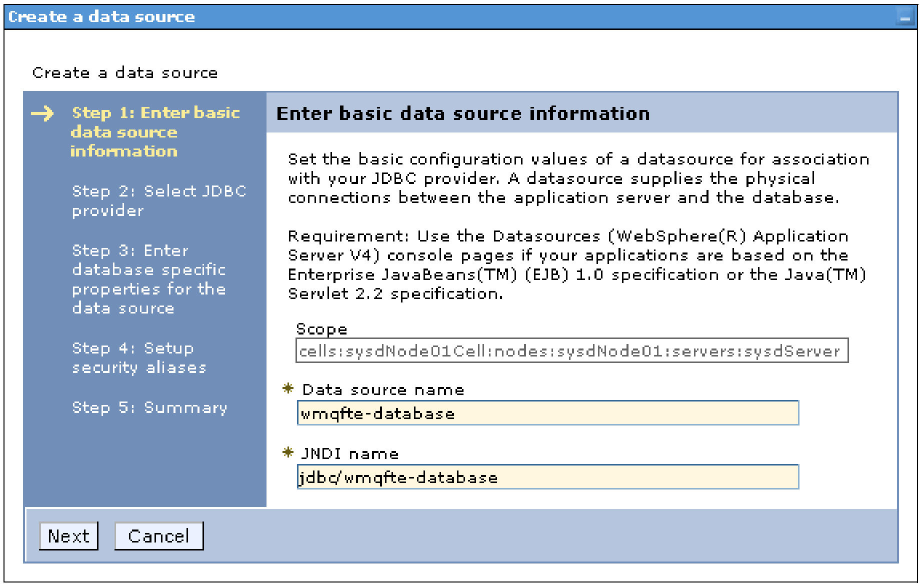
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

Node=sysdNode01, Server=sysdServer

Figure 5-15 JDBC provider scope

c. In Step 1, enter a data source name and a JNDI name (Figure 5-16):

- Data source name: `wmqfte-database`
- JNDI name: `jdbc/wmqfte-database`


Create a data source
Create a data source
Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
Step 5: Summary
Next
Cancel

Enter basic data source information

Set the basic configuration values of a datasource for association with your JDBC provider. A datasource supplies the physical connections between the application server and the database.

Requirement: Use the Datasources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans(TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.

Scope: cells:sysdNode01Cell:nodes:sysdNode01:servers:sysdServer

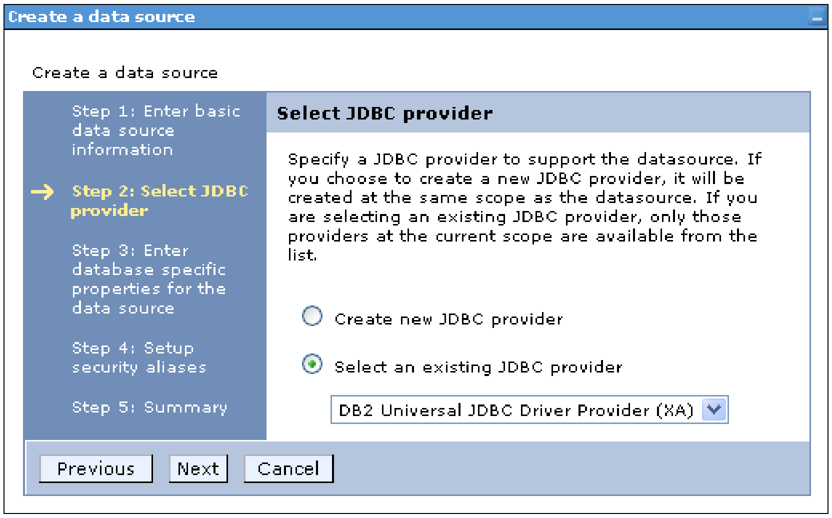
* Data source name: wmqfte-database

* JNDI name: jdbc/wmqfte-database

Figure 5-16 Data source wizard - Step 1

Click **Next**.

d. In Step 2, select the JDBC Provider that you created (Figure 5-17).


Create a data source
Create a data source
Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
Step 5: Summary
Previous
Next
Cancel

Select JDBC provider

Specify a JDBC provider to support the datasource. If you choose to create a new JDBC provider, it will be created at the same scope as the datasource. If you are selecting an existing JDBC provider, only those providers at the current scope are available from the list.

○ Create new JDBC provider

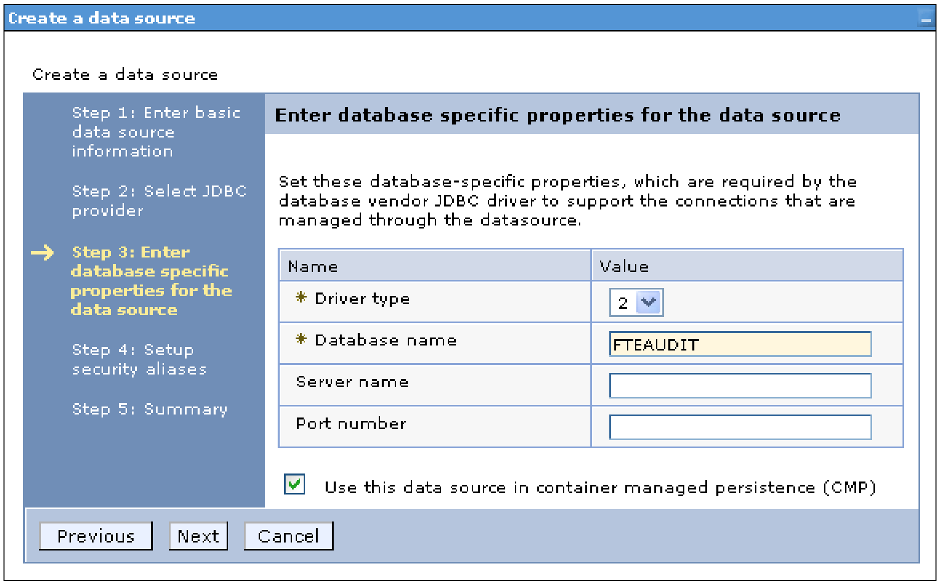
● Select an existing JDBC provider

DB2 Universal JDBC Driver Provider (XA)

Figure 5-17 Data source wizard - Step 2

Click **Next**.

e. In Step 3, enter the values required to access the database, including the driver type, database name, server name, and port number (Figure 5-18). These values can be obtained from the configuration of the database logger.


Create a data source
Create a data source
Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
Step 5: Summary
Previous
Next
Cancel

Enter database specific properties for the data source

Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
* Driver type	2
* Database name	FTEAUDIT
Server name	
Port number	

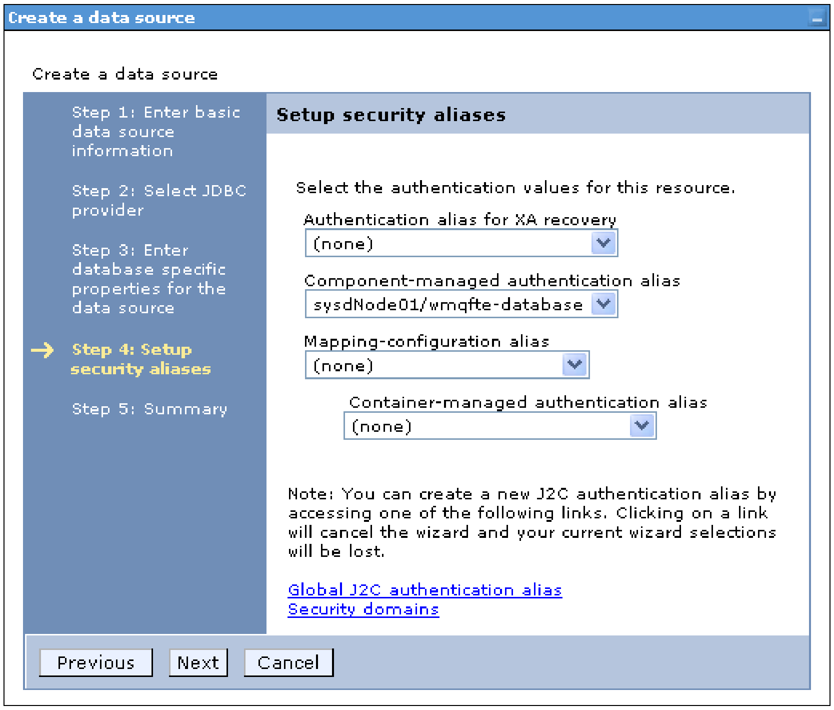
☑ Use this data source in container managed persistence (CMP)

Figure 5-18 Data source wizard - Step 3

For type 2 connections, the server name and port for the database are not needed.

Click **Next**.

f. In Step 4, supply the component-managed authentication alias that you created with the user ID required to access the database (Figure 5-19).


Create a data source
Create a data source
Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
Step 5: Summary
Previous
Next
Cancel

Setup security aliases

Select the authentication values for this resource.

Authentication alias for XA recovery
(none)

Component-managed authentication alias
sysdNode01/wmqfte-database

Mapping-configuration alias
(none)

Container-managed authentication alias
(none)

Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost.

Global J2C authentication alias
Security domains

Figure 5-19 Data source wizard - Step 4

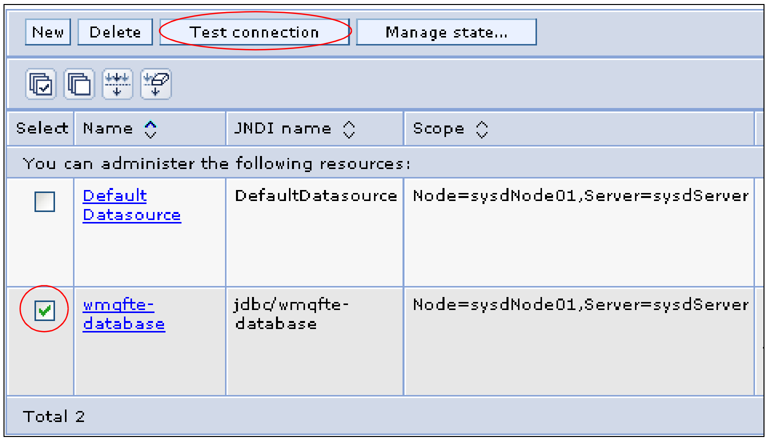
Click **Next**.

g. Click **Finish** on the summary page to create the data source.

4. To verify the configuration of the database:

   a. Select **Resources** → **JDBC** → **DataSources** from the administrative console navigation.

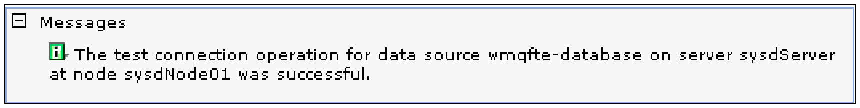
   b. Check the box next to your data source and click **Test connection** (Figure 5-20).



Select	Name	JNDI name	Scope
You can administer the following resources:			
□	Default Datasource	DefaultDatasource	Node=sysdNode01,Server=sysdServer
☑	wmqfte-database	jdbc/wmqfte-database	Node=sysdNode01,Server=sysdServer
Total 2			

Figure 5-20 Test the Database configuration

The results indicate that the connection was successful (Figure 5-21).


Messages

The test connection operation for data source wmqfte-database on server sysdServer at node sysdNode01 was successful.

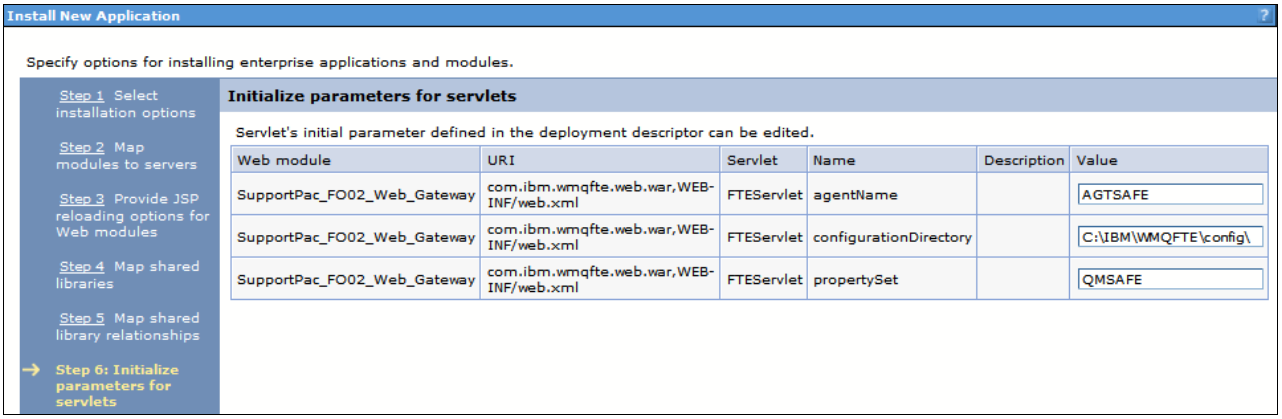
Figure 5-21 Successful data source test results

Deploying the Web Gateway application

Deploy the `fo02.ear` application file to the application server.

1. Select **Applications** → **New Application** from the administrative console.
2. Select **New Enterprise Application**.
3. Choose the **fo02.ear** file from the local file system and click **Next**.
4. Choose **Detailed Installation** and click **Next**.

5. Click the link to **Step 6: Initialize parameters for servlets** (Figure 5-22). Fill in the following information:
   – agentName: the agent running local to the Web Gateway
   – configurationDirectory: the configuration directory for your WebSphere MQ File Transfer installation, for example, `C:\IBM\WMQFTE\config\`
   – propertySet: the coordination queue manager


Install New Application
?
Step 1 Select installation options
Step 2 Map modules to servers
Step 3 Provide JSP reloading options for Web modules
Step 4 Map shared libraries
Step 5 Map shared library relationships
→ Step 6: Initialize parameters for servlets

Specify options for installing enterprise applications and modules.

Initialize parameters for servlets

Servlet's initial parameter defined in the deployment descriptor can be edited.

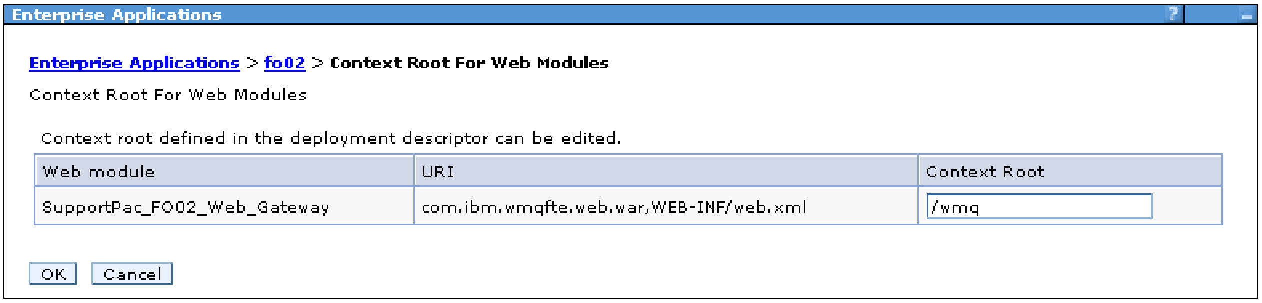
Web module	URI	Servlet	Name	Description	Value
SupportPac_FO02_Web_Gateway	com.ibm.wmqfte.web.war,WEB-INF/web.xml	FTEServlet	agentName		AGTSAFE
SupportPac_FO02_Web_Gateway	com.ibm.wmqfte.web.war,WEB-INF/web.xml	FTEServlet	configurationDirectory		C:\IBM\WMQFTE\config\
SupportPac_FO02_Web_Gateway	com.ibm.wmqfte.web.war,WEB-INF/web.xml	FTEServlet	propertySet		QMSAFE

Figure 5-22 Initialize parameters for servlets

6. Click the link to **Step 10: Map Virtual Hosts to Web Modules** and select the virtual host to be used (for example, `default_host`). Click **Next**.

7. For Step 11: Map context roots for web modules, enter `/wmq` for the context root (Figure 5-23).

   For this SupportPac, the context root must be /wmq.


Enterprise Applications
?
Enterprise Applications > fo02 > Context Root For Web Modules
OK
Cancel

Context Root For Web Modules

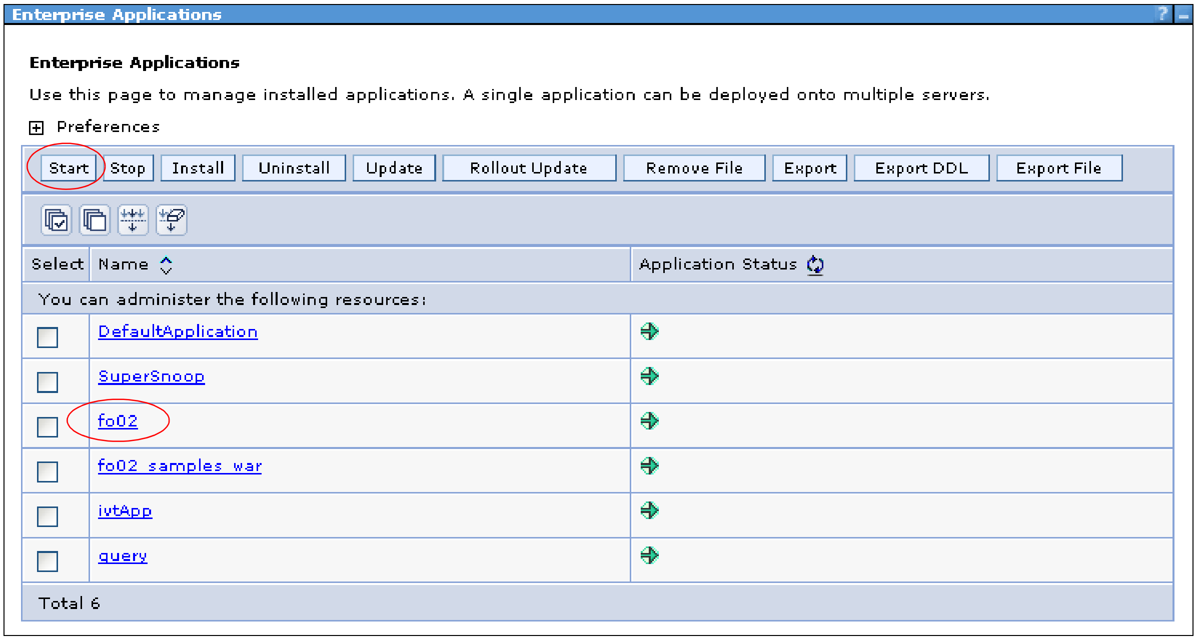
Context root defined in the deployment descriptor can be edited.

Web module	URI	Context Root
SupportPac_FO02_Web_Gateway	com.ibm.wmqfte.web.war,WEB-INF/web.xml	/wmq

Figure 5-23 Context root for Web Modules

8. Click the link to **Step 14 to review the summary** and click **Finish** to install the application.

9. To start the application, navigate to **Applications** → **Application Types** → **Enterprise Applications**. Select the check box next to fo02 and click **Start** (Figure 5-24).



Select	Name	Application Status
You can administer the following resources:		
□	DefaultApplication	➔
□	SuperSnoop	➔
□	fo02	➔
□	fo02_samples_war	➔
□	ivtApp	➔
□	query	➔
Total 6		

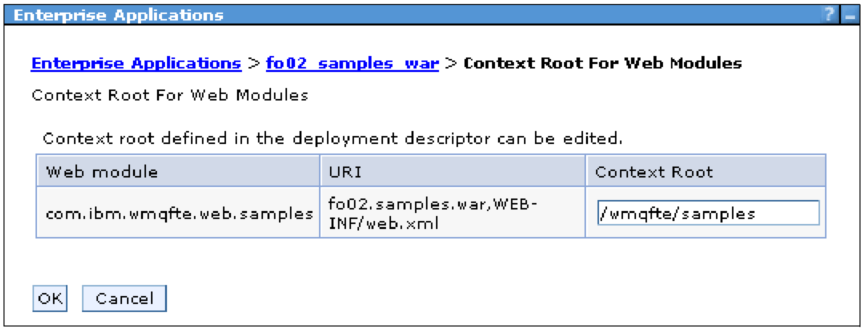
Figure 5-24 Enterprise Applications

The application is running when there is a green arrow next to the application name in the Application Status column.

Deploying the sample application

To configure the sample application to test or to see how the Web Gateway works:

1. Navigate to **Applications** → **New Application** → **New Enterprise Application**.
2. Browse the local file system to find where `fo02.samples.war` is located. Click **Next**.
3. Choose **Fast Path installation** and click **Next**.
4. Click the link to **Step 3: Map virtual hosts for web modules**. Select the virtual host (**default_host**) from the drop-down menu and click **Next**.
5. In Step 4: Map context roots for web modules, enter a unique context root for the sample application (Figure 5-25). Click **Next**.


Enterprise Applications
Enterprise Applications > fo02_samples_war > Context Root For Web Modules
Context Root For Web Modules
OK
Cancel

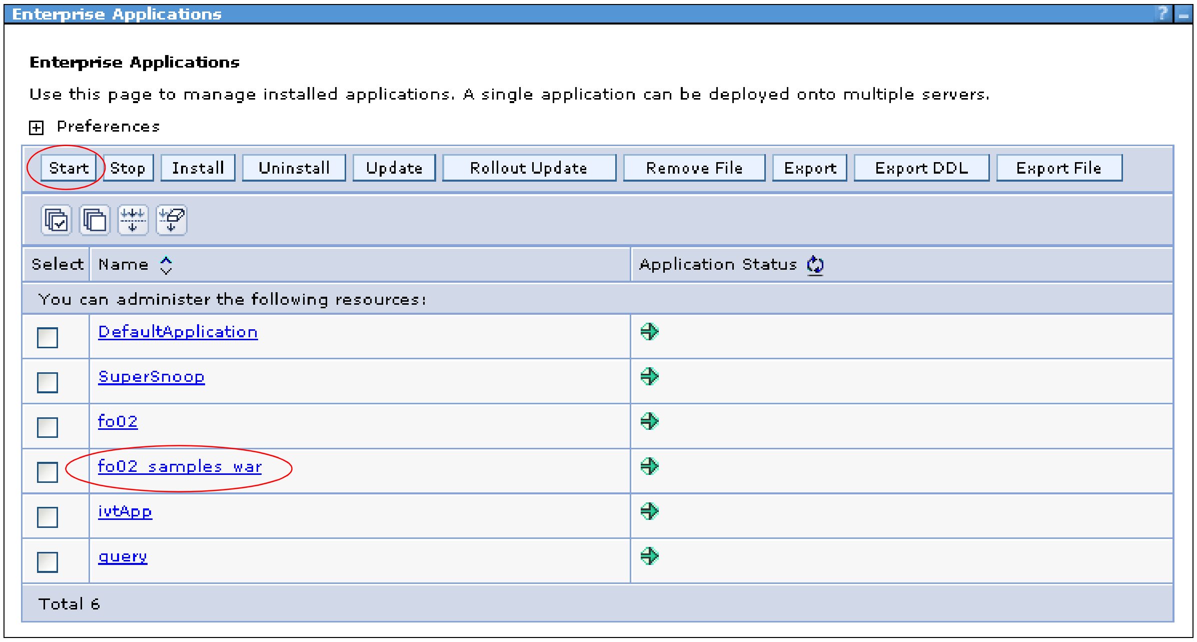
Context root defined in the deployment descriptor can be edited.

Web module	URI	Context Root
com.ibm.wmqfte.web.samples	fo02.samples.war,WEB-INF/web.xml	/wmqfte/samples

Figure 5-25 Sample application context root

6. Review the summary and click **Finish** to install the application.

7. To start the application, navigate to **Applications** → **Application Types** → **Enterprise Applications**. Select the check box next to fo02_samples_war and click **Start** (Figure 5-26).



Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

⊞ Preferences

Select	Name	Application Status
You can administer the following resources:		
□	DefaultApplication	➔
□	SuperSnoop	➔
□	fo02	➔
□	fo02_samples_war	➔
□	ivtApp	➔
□	query	➔
Total 6		

Figure 5-26 Start sample application

> **Updating the Web server plug-in configuration:** If the web gateway and sample application are installed after the HTTP server is configured and running with the Web server plug-in, the plug-in configuration file must be regenerated and copied to the HTTP server machine. In most circumstances when you are using an IBM HTTP Server, this is done automatically for you. However, if you find that you need to do this manually, you can perform these actions from the list of web servers in the administrative console for the application server.

5.4 Testing the scenario

To test the configuration of our scenario, we used the sample application provided in the FO02 SupportPac: WebSphere MQ File Transfer Edition - Web Gateway. The sample application creates a web page that can be accessed through a URL to upload files and see the status of the transfer.

To test the scenario use the following steps:

1. Open a web browser and enter the URL to access the sample application:

   `https://`*`yourHTTPServerHostName`*`/`*`yourContextRoot`*`/FileUpload.html`

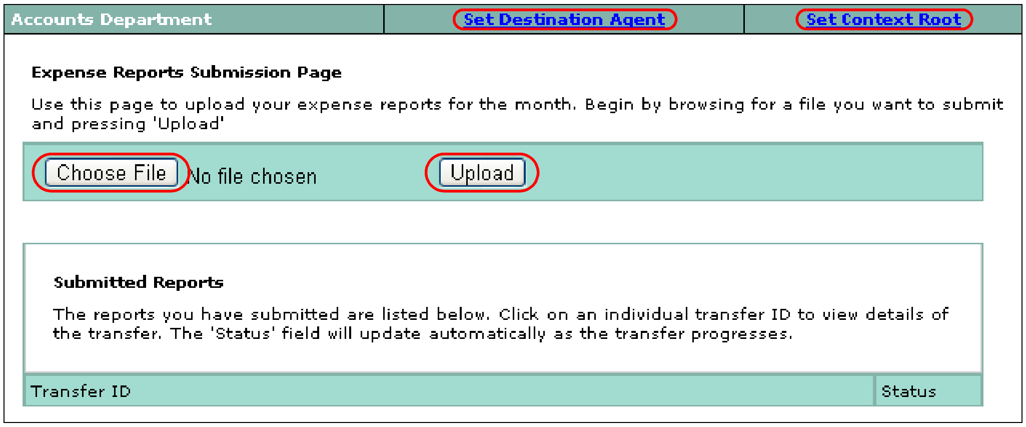
   In our system, the address is `https://sysd:9444/wmqfte/samples/FileUpload.html`.

   Note that the URL uses https and not standard http. This is to direct the request to a secure port for the file transfer.

2. Depending on the selected browser, you may encounter alert windows.

If you see a security alert regarding the certificate and asking whether you want to proceed, select the option allowing you to proceed (for example, **Yes** or **Proceed anyway**). What you see varies depending on the web browser.

3. You have reached the sample application when you see a panel similar to Figure 5-27 in your web browser.



Accounts Department | Set Destination Agent | Set Context Root

Expense Reports Submission Page

Use this page to upload your expense reports for the month. Begin by browsing for a file you want to submit and pressing 'Upload'

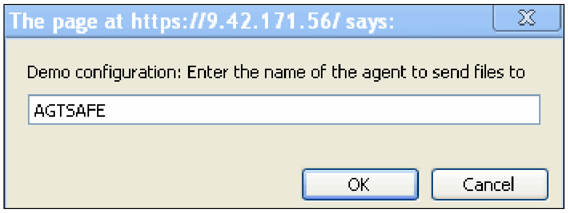
Submitted Reports

The reports you have submitted are listed below. Click on an individual transfer ID to view details of the transfer. The 'Status' field will update automatically as the transfer progresses.

Transfer ID	Status

Figure 5-27 Sample application

a. Click the **Set Destination Agent** link to specify the destination agent, AGTSAFE (Figure 5-28).



The page at https://9.42.171.56/ says:

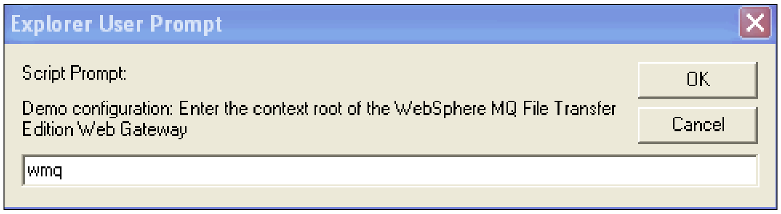
Demo configuration: Enter the name of the agent to send files to

AGTSAFE

Figure 5-28 Set destination agent prompt

Click **OK**.

b. Click **Set Context Root** to verify the context root for the Web Gateway application (Figure 5-29). For the SupportPac the context root must be wmq.



Explorer User Prompt

Script Prompt:

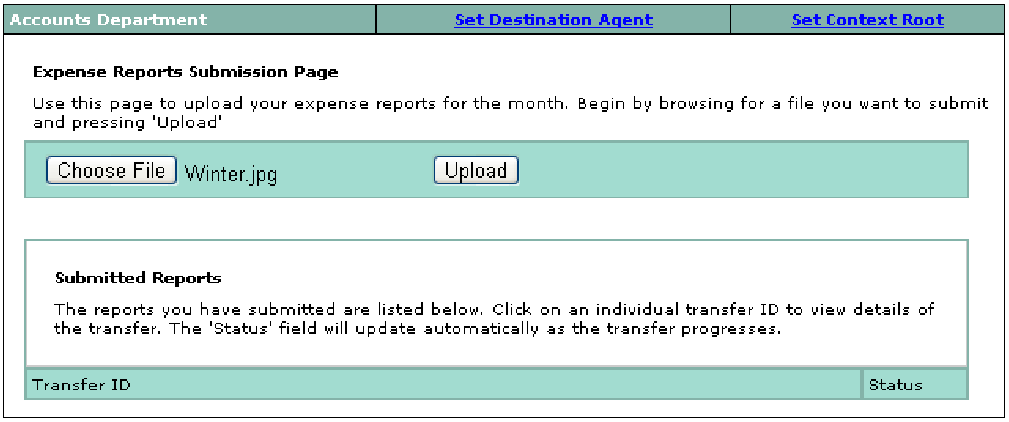
Demo configuration: Enter the context root of the WebSphere MQ File Transfer Edition Web Gateway

wmq

Figure 5-29 Context root prompt

Click **OK**.

c. Click **Choose File** to select the file to transfer (Figure 5-30 on page 73).



Expense Reports Submission Page

Use this page to upload your expense reports for the month. Begin by browsing for a file you want to submit and pressing 'Upload'

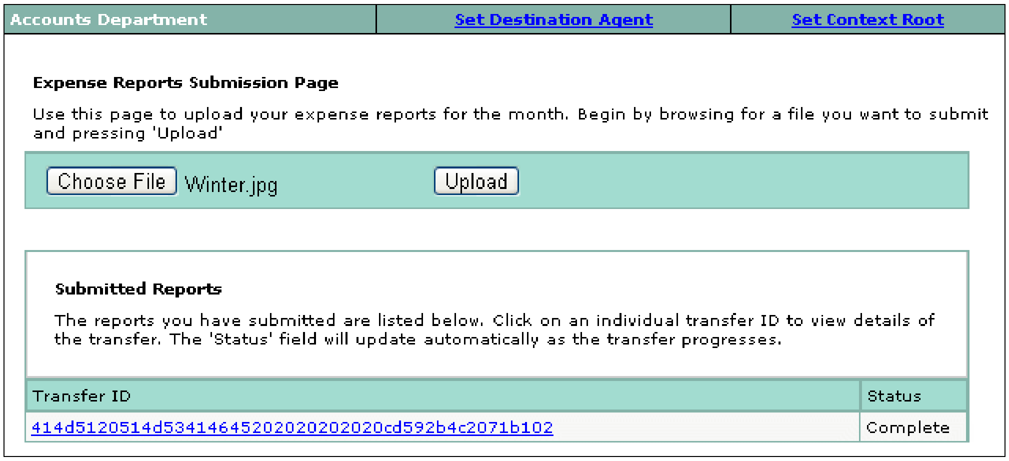
Submitted Reports

The reports you have submitted are listed below. Click on an individual transfer ID to view details of the transfer. The 'Status' field will update automatically as the transfer progresses.

Transfer ID	Status

Figure 5-30 File selected for transfer

d. Click **Upload** to transmit the file (Figure 5-31).



Expense Reports Submission Page

Use this page to upload your expense reports for the month. Begin by browsing for a file you want to submit and pressing 'Upload'

Submitted Reports

The reports you have submitted are listed below. Click on an individual transfer ID to view details of the transfer. The 'Status' field will update automatically as the transfer progresses.

Transfer ID	Status
414d5120514d53414645202020202020cd592b4c2071b102	Complete

Figure 5-31 Successful file transmission

4. To see the details of the complete transfer, click the link in the Transfer ID column of the Submitted Reports table. You will see the audit information stored as XML associated with the transfer.

   In certain browsers you may have to right-click and select **View page source** (Figure 5-32).

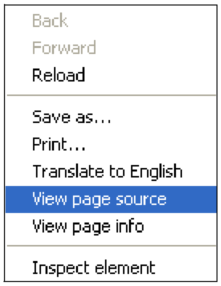
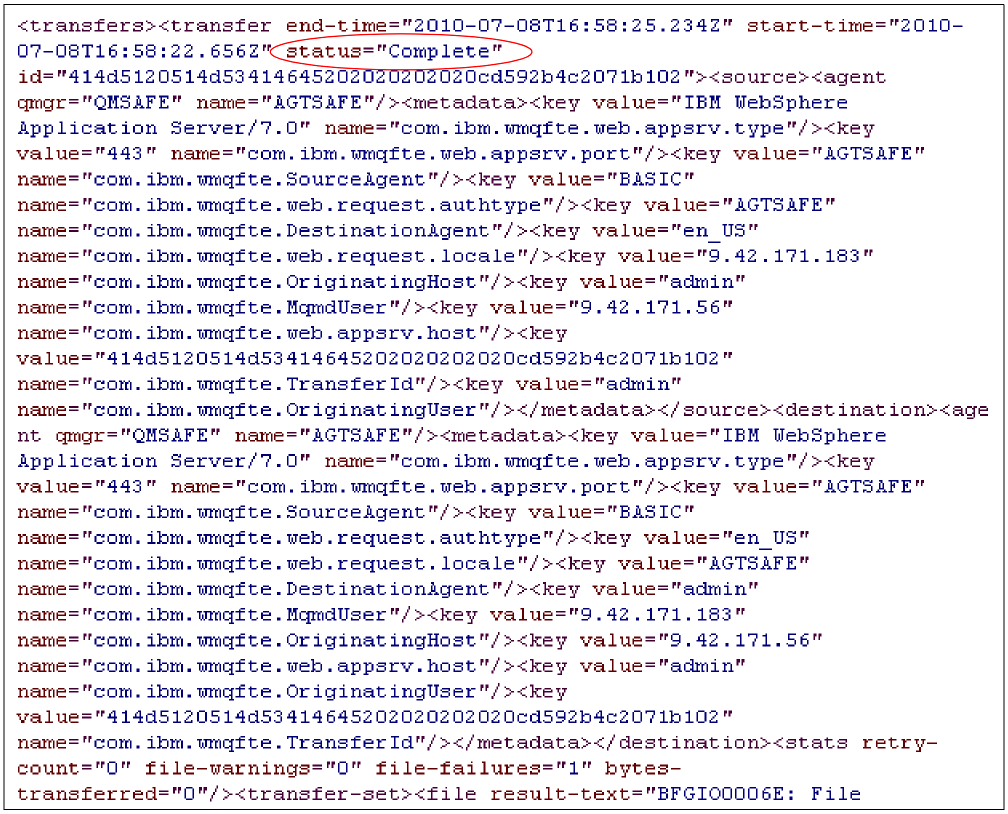


Figure 5-32 View page source option

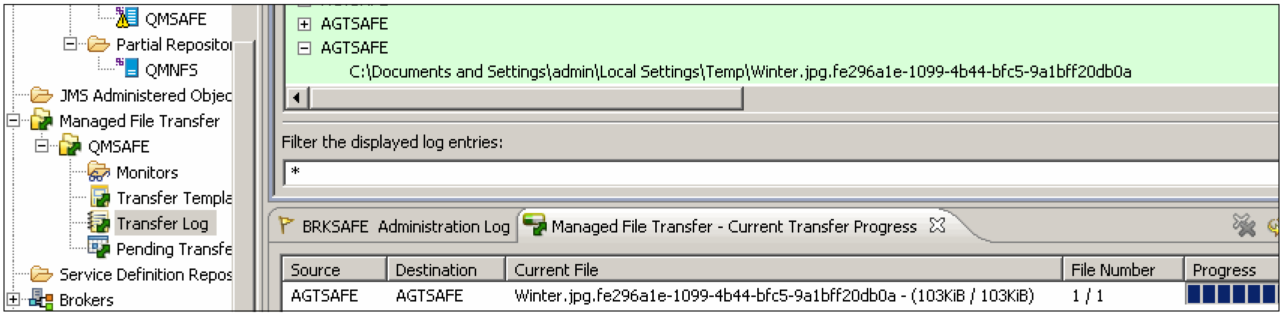
The results will resemble Figure 5-33. The Status=”Complete” message shows that the transfer completed.



```
<transfers><transfer end-time="2010-07-08T16:58:25.234Z" start-time="2010-
07-08T16:58:22.656Z" status="Complete"
id="414d5120514d534146452020202020cd592b4c2071b102"><source><agent
qmgr="QMSAFE" name="AGTSAFE"/><metadata><key value="IBM WebSphere
Application Server/7.0" name="com.ibm.wmqfte.web.appsrv.type"/><key
value="443" name="com.ibm.wmqfte.web.appsrv.port"/><key value="AGTSAFE"
name="com.ibm.wmqfte.SourceAgent"/><key value="BASIC"
name="com.ibm.wmqfte.web.request.authtype"/><key value="AGTSAFE"
name="com.ibm.wmqfte.DestinationAgent"/><key value="en_US"
name="com.ibm.wmqfte.web.request.locale"/><key value="9.42.171.183"
name="com.ibm.wmqfte.OriginatingHost"/><key value="admin"
name="com.ibm.wmqfte.MqmdUser"/><key value="9.42.171.56"
name="com.ibm.wmqfte.web.appsrv.host"/><key
value="414d5120514d534146452020202020cd592b4c2071b102"
name="com.ibm.wmqfte.TransferId"/><key value="admin"
name="com.ibm.wmqfte.OriginatingUser"/></metadata></source><destination><age
nt qmgr="QMSAFE" name="AGTSAFE"/><metadata><key value="IBM WebSphere
Application Server/7.0" name="com.ibm.wmqfte.web.appsrv.type"/><key
value="443" name="com.ibm.wmqfte.web.appsrv.port"/><key value="AGTSAFE"
name="com.ibm.wmqfte.SourceAgent"/><key value="BASIC"
name="com.ibm.wmqfte.web.request.authtype"/><key value="en_US"
name="com.ibm.wmqfte.web.request.locale"/><key value="AGTSAFE"
name="com.ibm.wmqfte.DestinationAgent"/><key value="admin"
name="com.ibm.wmqfte.MqmdUser"/><key value="9.42.171.183"
name="com.ibm.wmqfte.OriginatingHost"/><key value="9.42.171.56"
name="com.ibm.wmqfte.web.appsrv.host"/><key value="admin"
name="com.ibm.wmqfte.OriginatingUser"/><key
value="414d5120514d534146452020202020cd592b4c2071b102"
name="com.ibm.wmqfte.TransferId"/></metadata></destination><stats retry-
count="0" file-warnings="0" file-failures="1" bytes-
transferred="0"/><transfer-set><file result-text="BFGIO0006E: File
```

Figure 5-33 Sample application transfer ID XML

The transfer can also be viewed in the WebSphere MQ Explorer’s Transfer Log (Figure 5-34).



C:\Documents and Settings\admin\Local Settings\Temp\Winter.jpg.fe296a1e-1099-4b44-bfc5-9a1bff20db0a

Filter the displayed log entries: *

Source	Destination	Current File	File Number	Progress
AGTSAFE	AGTSAFE	Winter.jpg.fe296a1e-1099-4b44-bfc5-9a1bff20db0a - (103KiB / 103KiB)	1 / 1	■■■■■■

Figure 5-34 Transfer as seen in WebSphere MQ Explorer

5.5 Troubleshooting

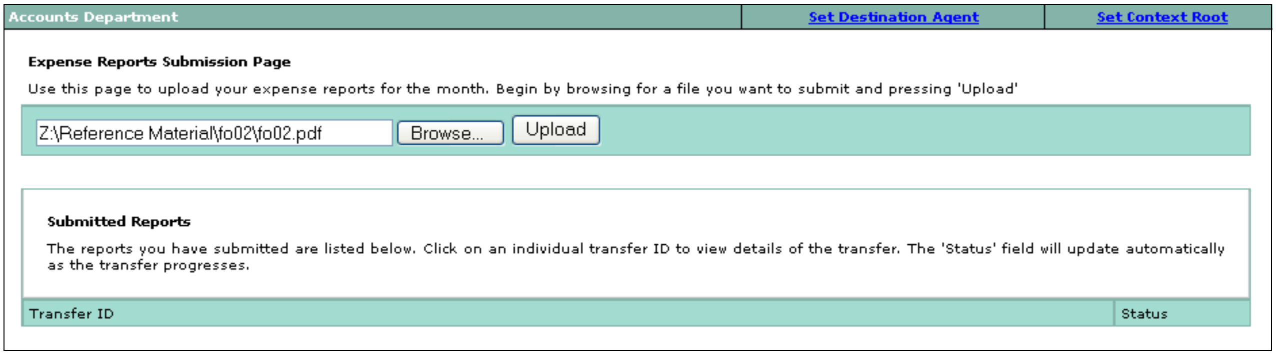
If things go wrong, there are simple steps to follow to determine where the problem might lie.

5.5.1 Previously working Web Gateway or sample application

There are two cases that fit this description, which we discuss in this section.

Nothing happens when you click Upload

If you click **Upload** in the sample application and nothing happens (Figure 5-35), there are a few items to check.



Accounts Department

Expense Reports Submission Page

Use this page to upload your expense reports for the month. Begin by browsing for a file you want to submit and pressing 'Upload'

Z:\Reference Material\fo02\fo02.pdf

Submitted Reports

The reports you have submitted are listed below. Click on an individual transfer ID to view details of the transfer. The 'Status' field will update automatically as the transfer progresses.

Transfer ID	Status

Figure 5-35 Sample application not responding when selecting Upload

Take the following steps:

1. Ping the agent to which the transfer is being sent and that the Web Gateway is using to send the file. In our scenario, this is AGTSAFE.

   – A successful ping looks like Example 5-6. This means that the agent is up and running.

   Example 5-6 Ping an agent successfully

   ```
   C:\Program Files\IBM\WMQFTE\bin>ftePingAgent AGTSAFE
   5655-U80, 5724-R10 Copyright IBM Corp.  2008, 2009.  ALL RIGH
   BFGCL0212I: Issuing ping request to agent AGTSAFE
   BFGCL0213I: agent AGTSAFE responded to ping in 2.359 seconds.
   ```

   – An unsuccessful ping looks like Example 5-7.

   Example 5-7 Ping an agent unsuccessfully

   ```
   C:\Program Files\IBM\WMQFTE\bin>ftePingAgent AGTSAFE
   5655-U80, 5724-R10 Copyright IBM Corp.  2008, 2009.  ALL RIGHTS RESERVED
   BFGCL0212I: Issuing ping request to agent AGTSAFE
   BFGCL0214I: agent AGTSAFE didn't respond to ping after 5 seconds.
   ```

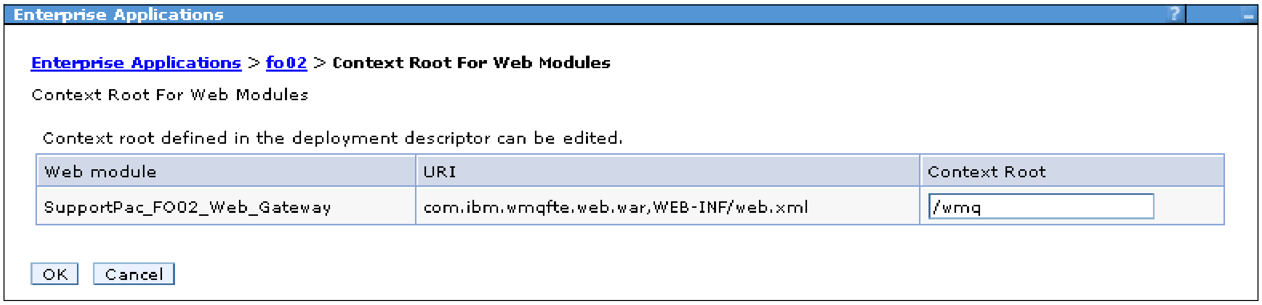
   If your ping is unsuccessful, then try starting the agent as shown in Example 5-8.

   Example 5-8 Command to start an agent

   ```
   C:\Program Files\IBM\WMQFTE\bin>fteStartAgent AGTSAFE
   5655-U80, 5724-R10 Copyright IBM Corp.  2008, 2009.  ALL RIGHTS RESERVED
   BFGCL0030I: The request to start agent 'AGTSAFE' on this machine has been
   submitted.
   BFGCL0031I: Agent log files located at:
   C:\IBM\WMQFTE\config\QMSAFE\agents\AGTSAFE
   ```

2. Make sure that the context root defined for the fo02 application in the application server, and the context root defined on the web page for the sample application, are both set to wmq.

   a. Navigate to the application's context root in the WebSphere Application Server's administrative console by expanding **Applications** → **Application Types** → **WebSphere enterprise applications**. Click **fo02** → **Context Root for Web modules**. This opens the panel shown in Figure 5-36. You can see that the context root is set to /wmq. If it is not, you can change it here by typing the context root and clicking **OK**.



Enterprise Applications

Enterprise Applications > fo02 > Context Root For Web Modules

Context Root For Web Modules

Context root defined in the deployment descriptor can be edited.

Web module	URI	Context Root
SupportPac_FO02_Web_Gateway	com.ibm.wmqfte.web.war,WEB-INF/web.xml	/wmq

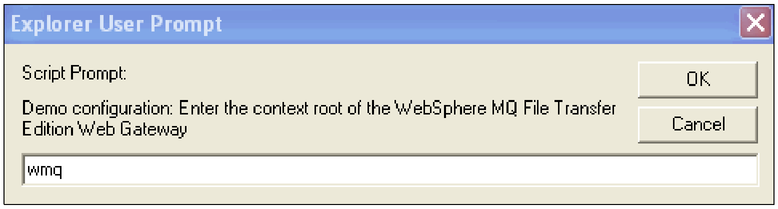
Figure 5-36 WebSphere Application Server fo02 application's context root

   b. Access the sample application from a browser:

   `https://sysd:9444/wmqfte/samples/FileUpload.html`

   From the first panel of the application, click **Set Context Root**.

   This opens the panel shown in Figure 5-37. You can set the context root for the gateway application here.



Explorer User Prompt

Script Prompt:

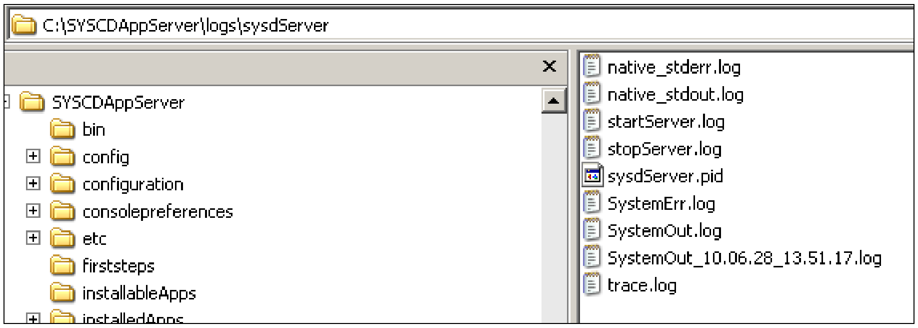
Demo configuration: Enter the context root of the WebSphere MQ File Transfer Edition Web Gateway

wmq

Figure 5-37 Sample application's context root

3. Check the WebSphere Application Server's logs to see whether there are any error messages (Figure 5-38). By default, the log is at:

   `<AppServerConfigDirectory>/logs/<appServer>/SystemOut.log`



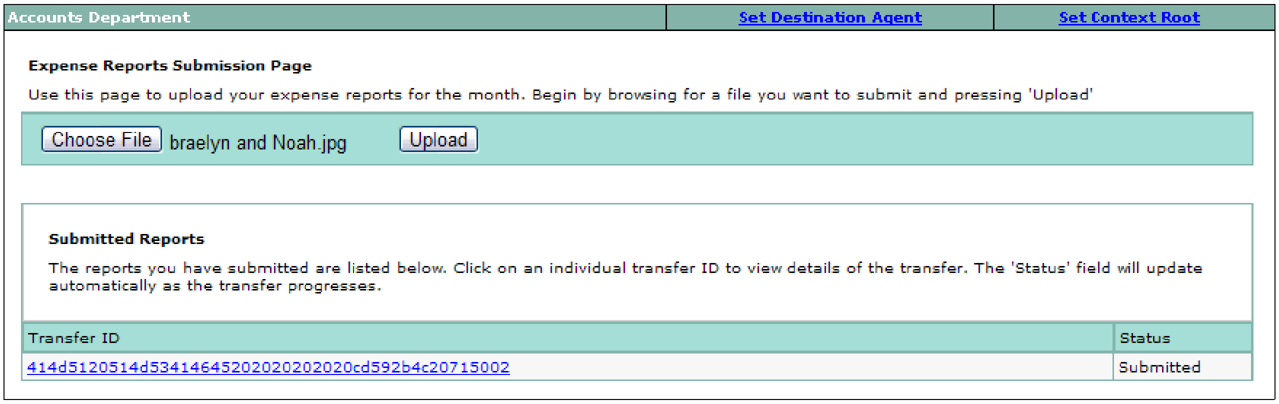
C:\SYSCDAppServer\logs\sysdServer

native_stderr.log
native_stdout.log
startServer.log
stopServer.log
sysdServer.pid
SystemErr.log
SystemOut.log
SystemOut_10.06.28_13.51.17.log
trace.log

Figure 5-38 WebSphere Application Server log files

The file transmission does not complete

If you transmit a file, but it never says that the file transmission completed (Figure 5-39), then this is an indication that there is a database logger issue.



Expense Reports Submission Page

Use this page to upload your expense reports for the month. Begin by browsing for a file you want to submit and pressing 'Upload'

Submitted Reports

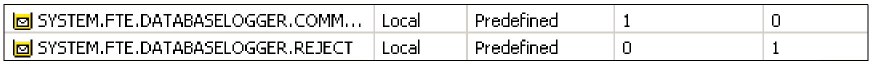
The reports you have submitted are listed below. Click on an individual transfer ID to view details of the transfer. The 'Status' field will update automatically as the transfer progresses.

Transfer ID	Status
414d5120514d53414645202020202020cd592b4c20715002	Submitted

Figure 5-39 Sample application transfer not showing complete

Take the following steps:

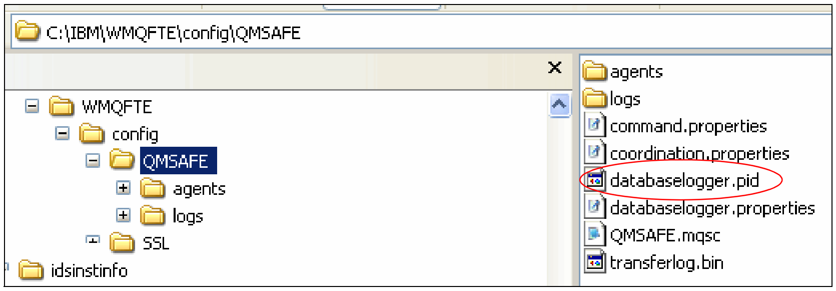
1. Check the database logger queues to ensure that the database logger is running. When the database logger is running, you will see a count of 1 on the input count in the first numerical column (Figure 5-40).



SYSTEM.FTE.DATABASELOGGER.COMM...	Local	Predefined	1	0
SYSTEM.FTE.DATABASELOGGER.REJECT	Local	Predefined	0	1

Figure 5-40 Database logger queues

You can also check in your configuration directory to make sure that the `databaselogger.pid` is shown (Figure 5-41). This `databaselogger.pid` indicates that the database logger is running.



C:\IBM\WMQFTE\config\QMSAFE

Figure 5-41 databaselogger.pid

If the database logger does not appear to be running, issue the **`fteStartDatabaseLogger`** command (Example A-17 on page 293).

2. Check the database logger logs to see whether there is an error (Figure 5-42).

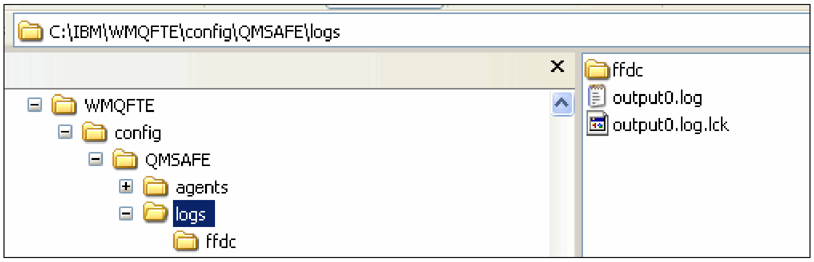


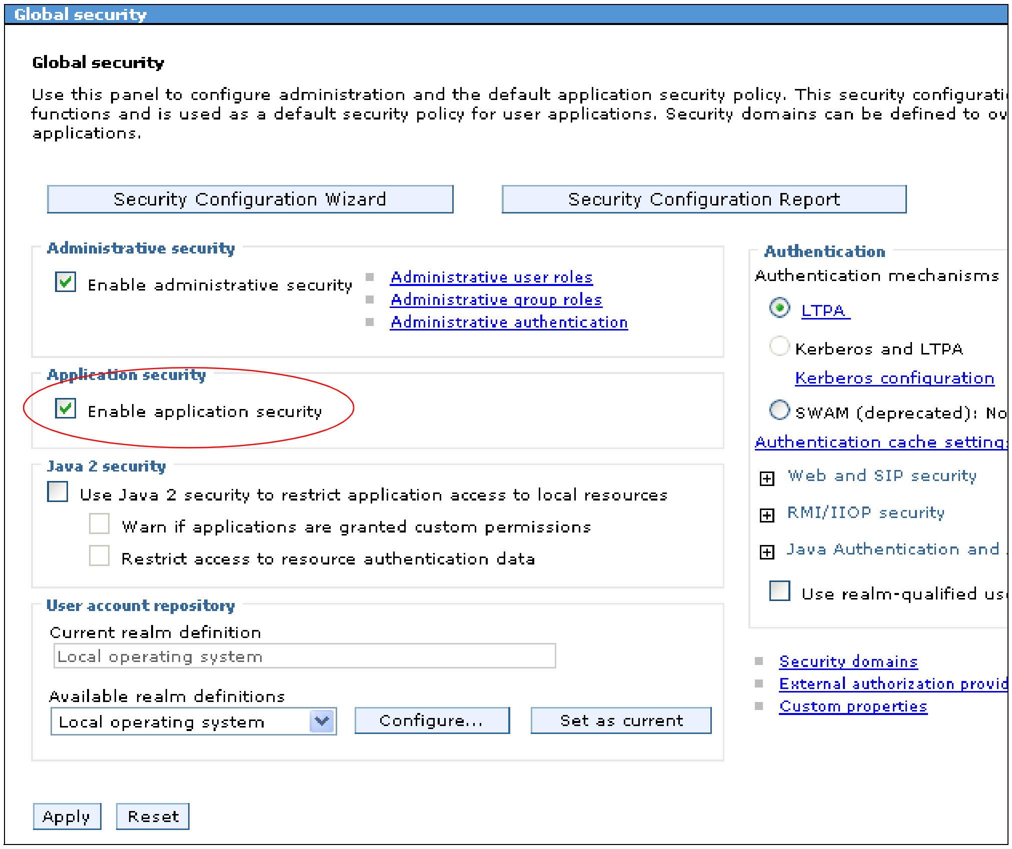
Figure 5-42 Database logger log file

5.5.2 Accessing the sample application and Web Gateway for the first time

If you are accessing the sample application and or Web Gateway for the first time, the troubleshooting tips in this section might be helpful, in addition to the tips in 5.5.1, “Previously working Web Gateway or sample application” on page 74.

If you have application security configured for your WebSphere Application Server, you must map a user ID to the Web Gateway roles:

1. To verify that you have application security turned on, view the WebSphere Application Server administrative console, expand **Security**, and select **Global Security**. If the box next to Application Security is checked (Figure 5-43), then application security is active.



Global security

Global security

Use this panel to configure administration and the default application security policy. This security configurati functions and is used as a default security policy for user applications. Security domains can be defined to ov applications.

Administrative security
☑ Enable administrative security

Application security
☑ Enable application security

Java 2 security
☐ Use Java 2 security to restrict application access to local resources
☐ Warn if applications are granted custom permissions
☐ Restrict access to resource authentication data

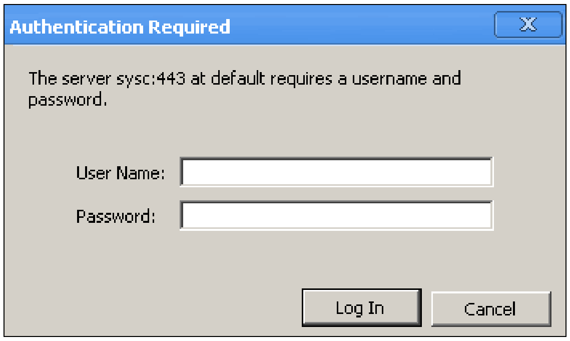
User account repository
Current realm definition: Local operating system
Available realm definitions: Local operating system

Authentication
Authentication mechanisms
● LTPA
○ Kerberos and LTPA
○ SWAM (deprecated): No
☐ Use realm-qualified us

Security Configuration Wizard
Security Configuration Report
Administrative user roles
Administrative group roles
Administrative authentication
Kerberos configuration
Authentication cache setting
Web and SIP security
RMI/IIOP security
Java Authentication and
Configure...
Set as current
Security domains
External authorization provid
Custom properties
Apply
Reset

Figure 5-43 Global security with application security enabled

You can also verify that the application security is turned on by accessing the sample application through the URL. If you receive a prompt asking for a user ID and password (Figure 5-44) while following the steps in 5.4, “Testing the scenario” on page 71, then application security is turned on.



Authentication Required

The server sysc:443 at default requires a username and password.

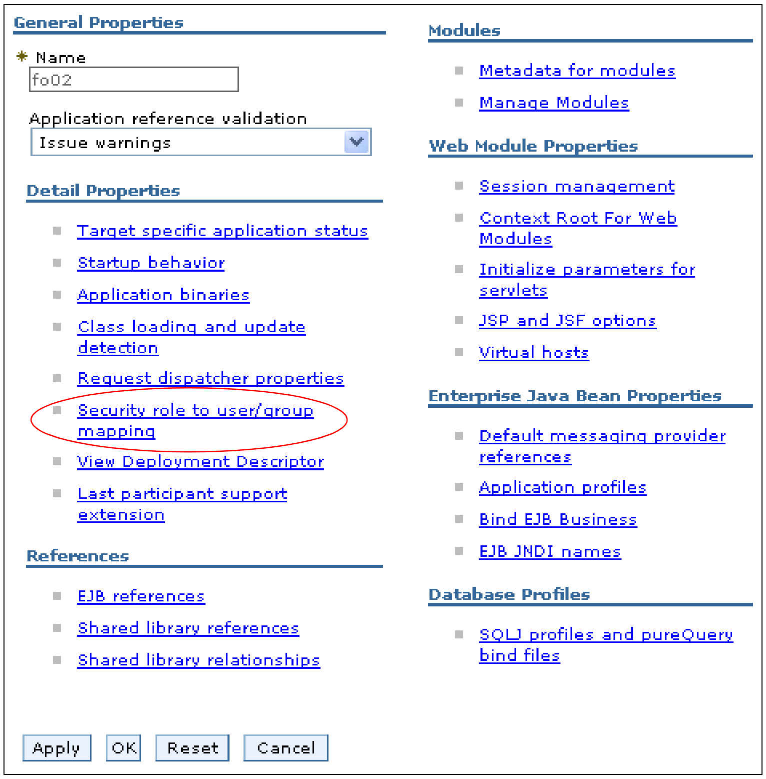
User Name:

Password:

Figure 5-44 Sample application prompting for user name and password

If application security is active, proceed to the following steps.

2. To map a user ID to the sample application, in the administrative console navigate to **Applications** → **Application Types** → **WebSphere Enterprise Applications**. Click **fo02**. Select **Security role to user/group mapping** (Figure 5-45).



General Properties

* Name: fo02

Application reference validation: Issue warnings

Detail Properties

- Target specific application status
- Startup behavior
- Application binaries
- Class loading and update detection
- Request dispatcher properties
- Security role to user/group mapping
- View Deployment Descriptor
- Last participant support extension

References

- EJB references
- Shared library references
- Shared library relationships

Modules

- Metadata for modules
- Manage Modules

Web Module Properties

- Session management
- Context Root For Web Modules
- Initialize parameters for servlets
- JSP and JSF options
- Virtual hosts

Enterprise Java Bean Properties

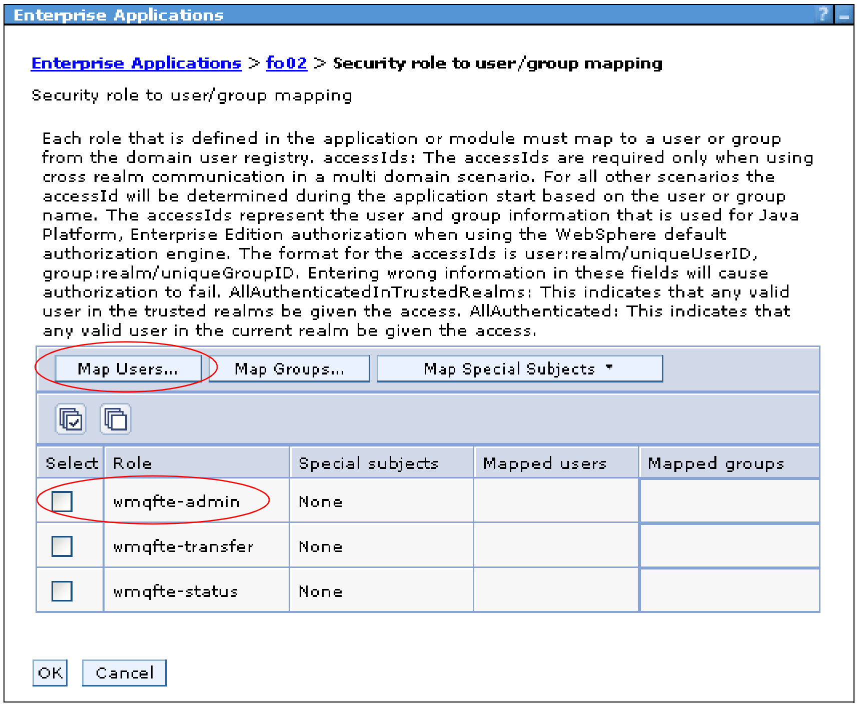
- Default messaging provider references
- Application profiles
- Bind EJB Business
- EJB JNDI names

Database Profiles

- SQLJ profiles and pureQuery bind files

Figure 5-45 Web gateway fo02 general properties in administrative console

3. Check the box next to wmqfte-admin and click **Map users** (Figure 5-46).



Enterprise Applications > fo02 > Security role to user/group mapping

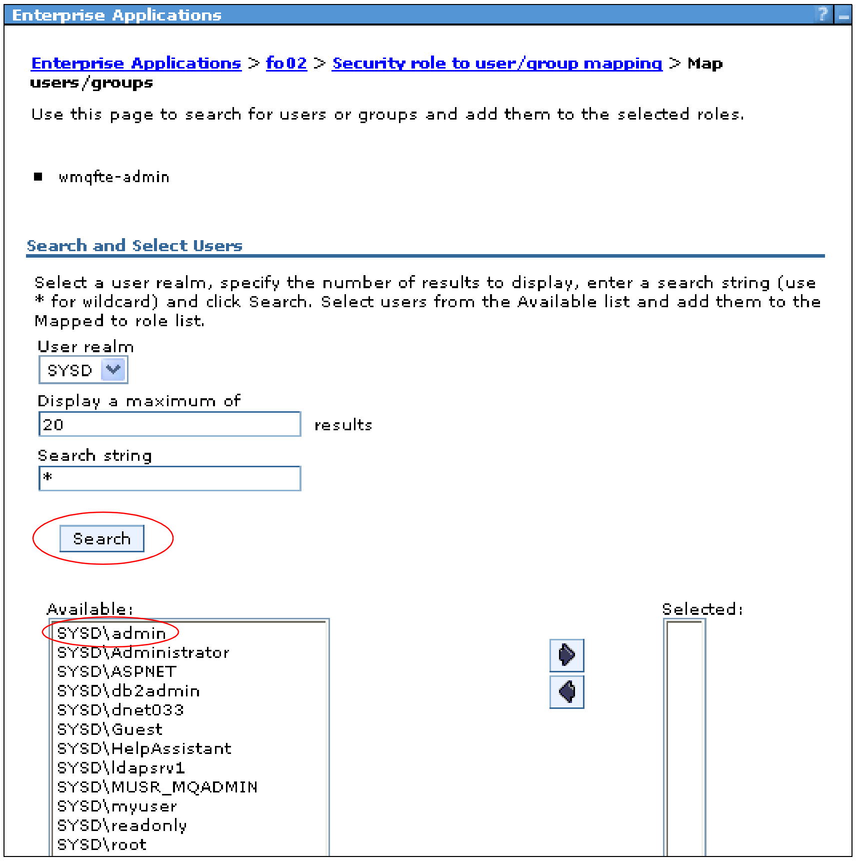
Security role to user/group mapping

Each role that is defined in the application or module must map to a user or group from the domain user registry. accessIds: The accessIds are required only when using cross realm communication in a multi domain scenario. For all other scenarios the accessId will be determined during the application start based on the user or group name. The accessIds represent the user and group information that is used for Java Platform, Enterprise Edition authorization when using the WebSphere default authorization engine. The format for the accessIds is user:realm/uniqueUserID, group:realm/uniqueGroupID. Entering wrong information in these fields will cause authorization to fail. AllAuthenticatedInTrustedRealms: This indicates that any valid user in the trusted realms be given the access. AllAuthenticated: This indicates that any valid user in the current realm be given the access.

Select	Role	Special subjects	Mapped users	Mapped groups
□	wmqfte-admin	None		
□	wmqfte-transfer	None		
□	wmqfte-status	None		

Figure 5-46 Mapping users to wmqfte-admin

4. Click **Search** to see all of the user IDs available on the system (Figure 5-47).



Enterprise Applications > fo02 > Security role to user/group mapping > Map users/groups

Use this page to search for users or groups and add them to the selected roles.

- wmqfte-admin

Search and Select Users

Select a user realm, specify the number of results to display, enter a search string (use * for wildcard) and click Search. Select users from the Available list and add them to the Mapped to role list.

User realm: SYSD

Display a maximum of: 20 results

Search string: *

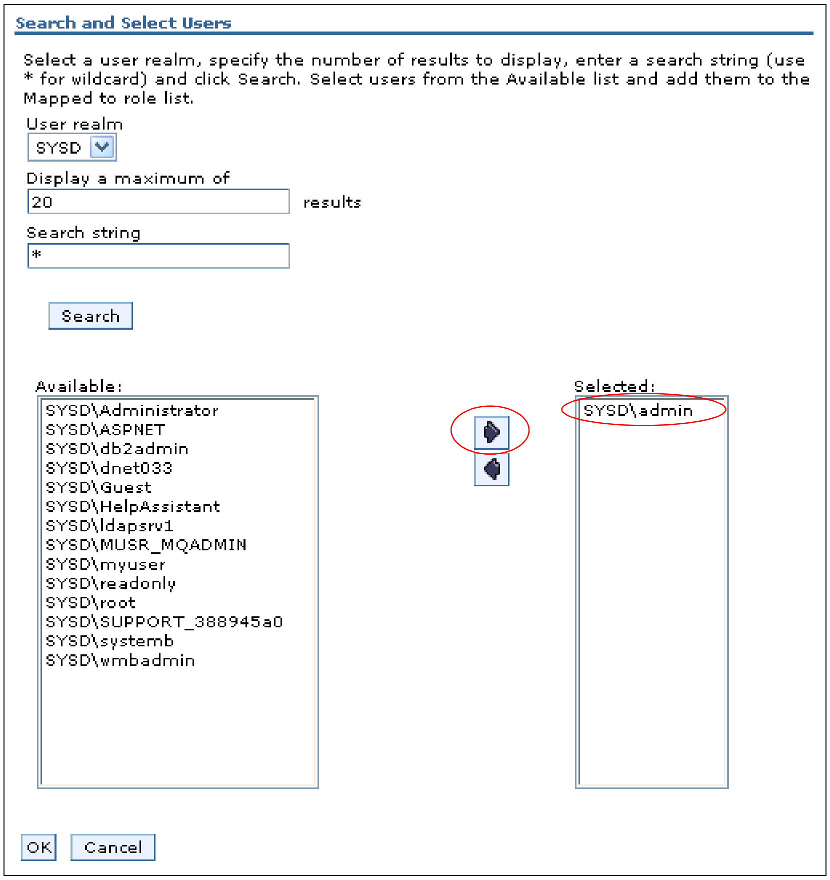
Available:

SYSD\admin
SYSD\Administrator
SYSD\ASPNET
SYSD\db2admin
SYSD\dnet033
SYSD\Guest
SYSD\HelpAssistant
SYSD\ldapsrv1
SYSD\MUSR_MQADMIN
SYSD\myuser
SYSD\readonly
SYSD\root

Selected:

Figure 5-47 Search the local system for user IDs

5. Select the user ID that you want to use and the arrow pointing to the right to move the user ID to the selected column (Figure 5-48). Click **OK**.



Search and Select Users

Select a user realm, specify the number of results to display, enter a search string (use * for wildcard) and click Search. Select users from the Available list and add them to the Mapped to role list.

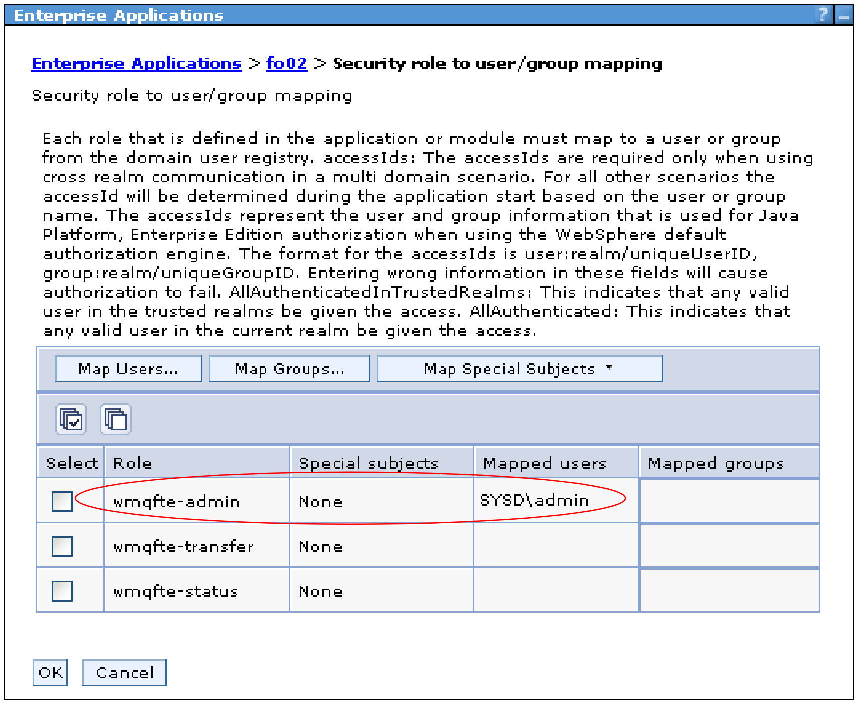
User realm: SYSD

Display a maximum of: 20 results

Search string: *

Figure 5-48 Moved admin user ID to the selected column

6. Review your choices on the Security role to user/group mapping page and click **OK** (Figure 5-49).



Enterprise Applications > fo02 > **Security role to user/group mapping**

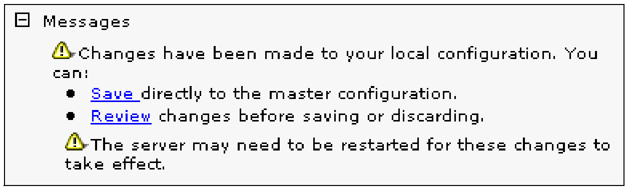
Security role to user/group mapping

Each role that is defined in the application or module must map to a user or group from the domain user registry. accessIds: The accessIds are required only when using cross realm communication in a multi domain scenario. For all other scenarios the accessId will be determined during the application start based on the user or group name. The accessIds represent the user and group information that is used for Java Platform, Enterprise Edition authorization when using the WebSphere default authorization engine. The format for the accessIds is user:realm/uniqueUserID, group:realm/uniqueGroupID. Entering wrong information in these fields will cause authorization to fail. AllAuthenticatedInTrustedRealms: This indicates that any valid user in the trusted realms be given the access. AllAuthenticated: This indicates that any valid user in the current realm be given the access.

Select	Role	Special subjects	Mapped users	Mapped groups
□	wmqfte-admin	None	SYSD\admin	
□	wmqfte-transfer	None		
□	wmqfte-status	None		

Figure 5-49 Review mapped users for wmqfte-admin

7. Save the changes to the master configuration by clicking the **Save** link (Figure 5-50).



⊟ Messages

⚠ Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

Figure 5-50 Save changes to master configuration

8. Restart the application.

9. In your web browser, go back to the sample application URL and follow the steps in 5.4, “Testing the scenario” on page 71. This time, when the pop-up to enter a user name and password displays, enter the user name that you mapped in the previous steps and its password. You are now able to use the sample application with application security turned on.

6

Using FTP/SFTP with WebSphere MQ File Transfer Edition

In this chapter we discuss the challenge of integrating enterprises that use different protocols for file transfer. The scenario illustrates how WebSphere MQ File Transfer Edition can bridge to protocols, such as File Transfer Protocol (FTP) or Secure File Transfer Protocol (SFTP), using a protocol bridge.

> **Terminology:** This scenario applies to FTP and SFTP servers and protocols. For simplicity, we use the term SFTP to imply support for both.

6.1 Scenario overview

Many companies today use standard protocols such as FTP and SFTP for file transfer to and from their external partners. These file transfers are often not typical business-to-business scenarios, which have enhanced partner management requirements. Using FTP or SFTP provides a simple-to-use and low-cost platform for file exchange. But because of the shortcomings of these protocols, as discussed in "Challenges surrounding FTP in a multi-enterprise file transfer" on page 6, companies often have to provide significant effort to make their file transfers reliable and managed.

This chapter illustrates how WebSphere MQ File Transfer Edition can integrate with external partners that use FTP or SFTP. This integration uses a bridge agent, a specific type of WebSphere MQ File Transfer Edition agent, that uses SFTP as the transport protocol rather than the WebSphere MQ transport protocol. The bridge agent acts a the bridge between an SFTP server on one side and the MQ FTE backbone on the other side, leveraging the WebSphere MQ File Transfer Edition features to enable managed file transfer over the internet.

With a protocol bridge, the following actions can be performed:

- Files can be transferred from the MQ FTE backbone network to an SFTP server.
- Files can be retrieved from an SFTP server to an MQ FTE backbone network.

The scenario in this chapter shows how a bridge agent is created and configured to connect to an external partner to send and retrieve files over the internet.

6.1.1 Appropriate use

The bridge agent can be configured to use either FTP or SFTP as the underlying transport protocol. The agent acts as a client and connects to an FTP or SFTP server. In our scenario the bridge agent is configured for SFTP-based communication. All of the file transfers, inbound and outbound, are initiated by the bridge agent. It is not possible to initiate a file transfer from the SFTP server. This provides a high level of security because an external partner does not originate the connection into the enterprise.

The bridge agent requires a local queue manager and therefore is created and configured in binding mode to its agent queue manager. The bridge agent cannot store the retrieved files onto or read files from a local file system. This means that an additional agent is required in the WebSphere MQ File Transfer Edition network to act as the source or target agent for the local file system.

To transfer files, the bridge agent must have the appropriate permission to connect to the remote SFTP server. The access permissions to connect to the server can be configured based either on user ID/password credentials or a public/private key. You can find a detailed description of how to configure the protocol bridge agent credentials in 6.2.8, "Creating and configuring the bridge agent" on page 96. Furthermore, the bridge agent user must have the appropriate permissions to create and delete files on the SFTP server file system. For a detailed description of how to define these access permissions, see the documentation of the specific SFTP server that you are using.

6.1.2 Business value

Using the bridge agent allows for the easy and cost-efficient integration of an existing FTP or SFTP network to a managed network based on WebSphere MQ File Transfer Edition. It is not necessary to install WebSphere MQ File Transfer Edition code on the SFTP network, so there are no additional software costs to connect to external partners who are using SFTP today.

The bridge agent enhances SFTP-based file transfers with WebSphere MQ File Transfer Edition managed file transfer capabilities, such as:

- Reliable and assured file delivery.
- State management and automated restart of interrupted file transfers.
- File transfers are executed in multi-threading mode, meaning that multiple files can be sent and received concurrently.
- Files of unlimited size can be transferred.
- File transfers can be automated by setting up schedules or event-based trigger mechanisms.
- The status of the file transfers can be monitored in the WebSphere MQ File Transfer Edition Explorer.
- File transfers can be recorded in detail in the WebSphere MQ File Transfer Edition logging database for reporting and auditing purposes.

6.2 Scenario details

In these scenarios we highlight the fact that multiple enterprises can exchange files with each other by connecting an MQ FTE backbone with non-WebSphere MQ File Transfer Edition networks using a standard transport protocol such as FTP or SFTP in between. We show how to configure and set up a bridge agent that acts as the bridge between an SFTP Server outside of our enterprise and the MQ FTE backbone network within the enterprise.

We show the use of the bridge agent in two scenarios that differ slightly in the way in which the infrastructure is set up in the enterprise. Common to both scenarios is that all WebSphere MQ File Transfer Edition components, including the bridge agent, reside in the protected network, and that the external partner side has an SFTP server installed.

In the first scenario, we have an SFTP server and a local file system in the demilitarized zone (DMZ). Files are transferred between the SFTP servers on the partner side and in the DMZ over the internet. The files can be pushed or pulled in both directions, depending on agreements in place between the companies. The important thing to note with this topology is that the files are persisted in the local file system in the DMZ as they are retrieved from the partner and as they are sent out. This can have an impact with regard to compliance with security guidelines. We discuss this issue further in 6.2.5, “Security” on page 94.

The second scenario is similar to the first scenario, but in this case, traffic flows through the DMZ without stopping. The bridge agent (AGTSFTP) in the protected network connects directly to the SFTP server on the external partner side and initiates the file transfers in both directions. The files from the partner are transferred directly into the protected network. Because the bridge agent acts as an SFTP client, it is not possible for the SFTP server to which the agent is connected to initiate a transfer and send files to the agent. Because all file transfers are initiated from within the protected network, no inbound ports have to be opened in the firewall for incoming requests from external parties.

Which scenario you use depends on your relationship with the partner and how you agree to exchange and secure files. If the enterprise can initiate the file transfers to and from the partner, the second scenario might be preferable because it does not require that you open inbound ports in the firewall, and it provides real end-to-end managed file transfer.

To automate the inbound file transfers into the protected network (files received either from the external partner or from the temporary file store in the DMZ), we must define a scheduled file transfer for the bridge agent to check the file system to which it is connected on a regular basis for relevant files.

The outbound file transfers (from the file system in the protected network to the external partner or to the temporary file store in the DMZ) are initiated by the AGTSAFE agent. For automation purposes, we define a resource monitor for this agent. The resource monitor polls the file system for given events (the existence of new or updated files) and starts the file transfer.

We use WebSphere MQ Explorer with the WebSphere MQ File Transfer Edition Explorer plug-in in our scenarios to monitor the status of our file transfers.

6.2.1 Components

Figure 6-1 shows a high-level overview of all the components used in the scenarios in this chapter.

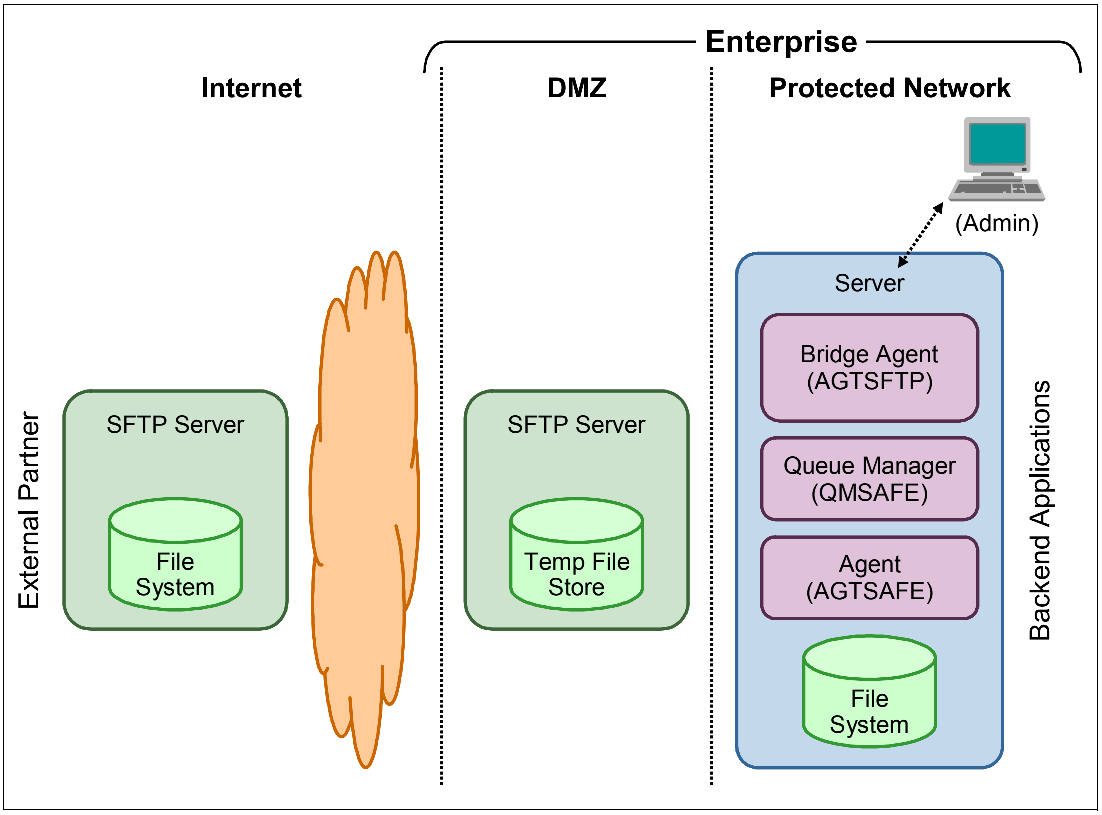


Figure 6-1 Scenario components

WebSphere MQ Queue Manager (QMSAFE)

Queue manager QMSAFE is the coordination and command queue manager for the MQ FTE backbone network. The coordination queue manager publishes status messages received from the agents. QMSAFE is also the agent queue manager for the AGTSAFE and AGTSFTP agents.

WebSphere MQ File Transfer EditionServer Agent (AGTSAFE)

AGTSAFE is the WebSphere MQ File Transfer Edition agent that connects to the local queue manager QMSAFE in bindings mode. This type of agent is referred to as a server agent. AGTSAFE reads files from and writes files to the local file system. This agent is the target agent for the bridge agent for inbound scenarios and the source agent for outbound scenarios.

WebSphere MQ File Transfer Editionbridge agent (AGTSFTP)

AGTSFTP is a bridge agent. It connects to the local queue manager QMSAFE in bindings mode. AGTSFTP connects to an SFTP server to send and retrieve files. Because this agent cannot store files to or read files from the local file system, it connects to AGTSAFE for file transfers. AGTSAFE has access to the file system.

WebSphere MQ Explorer

The WebSphere MQ Explorer is an Eclipse-based administration tool, shipped with WebSphere MQ. It is used to view and administer WebSphere MQ queue managers and queue manager objects, such as queues, channels, and topics.

WebSphere MQ File Transfer Edition Explorer

The WebSphere MQ File Transfer Edition Explorer is a plug-in to the WebSphere MQ Explorer. It is used to schedule file transfer requests and view the status of current file transfers. The tool includes a Transfer Log view, which subscribes to the coordination queue manager for the audit information. This view shows information about every file transfer that occurs in a given topology.

SFTP server

The OpenSSH server daemon on Linux is used as the SFTP server on the external partner side and the SFTP server in the DMZ in our scenarios.

6.2.2 File transfers to an SFTP server in the DMZ

In this scenario we have an SFTP server in the DMZ between the internet and the protected network. Files are exchanged between the SFTP server in the DMZ and the SFTP server on the external partner side. Incoming files sent by the external partner and outgoing files sent by the AGTSFTP agent are stored intermediately in the temporary file store.

> **Managed file transfer capabilities:** It is likely that this topology will be used by many companies because it supports the concept of using a DMZ in between the internet and the protected network, making the topology compliant with commonly held security rules and guidelines. However, the managed file transfer capabilities of WebSphere MQ File Transfer Edition are only applied when the AGTSFTP agent is involved in moving the data between the temporary file store and the protected network. The file transfers between the SFTP servers in the DMZ and on the external partner side are based on the standard SFTP protocol, and therefore additional effort is required for the implementation of managed file transfer capabilities such as automation, monitoring, and error handling.

Scenario inbound flow

Figure 6-2 shows the inbound flow for this scenario.

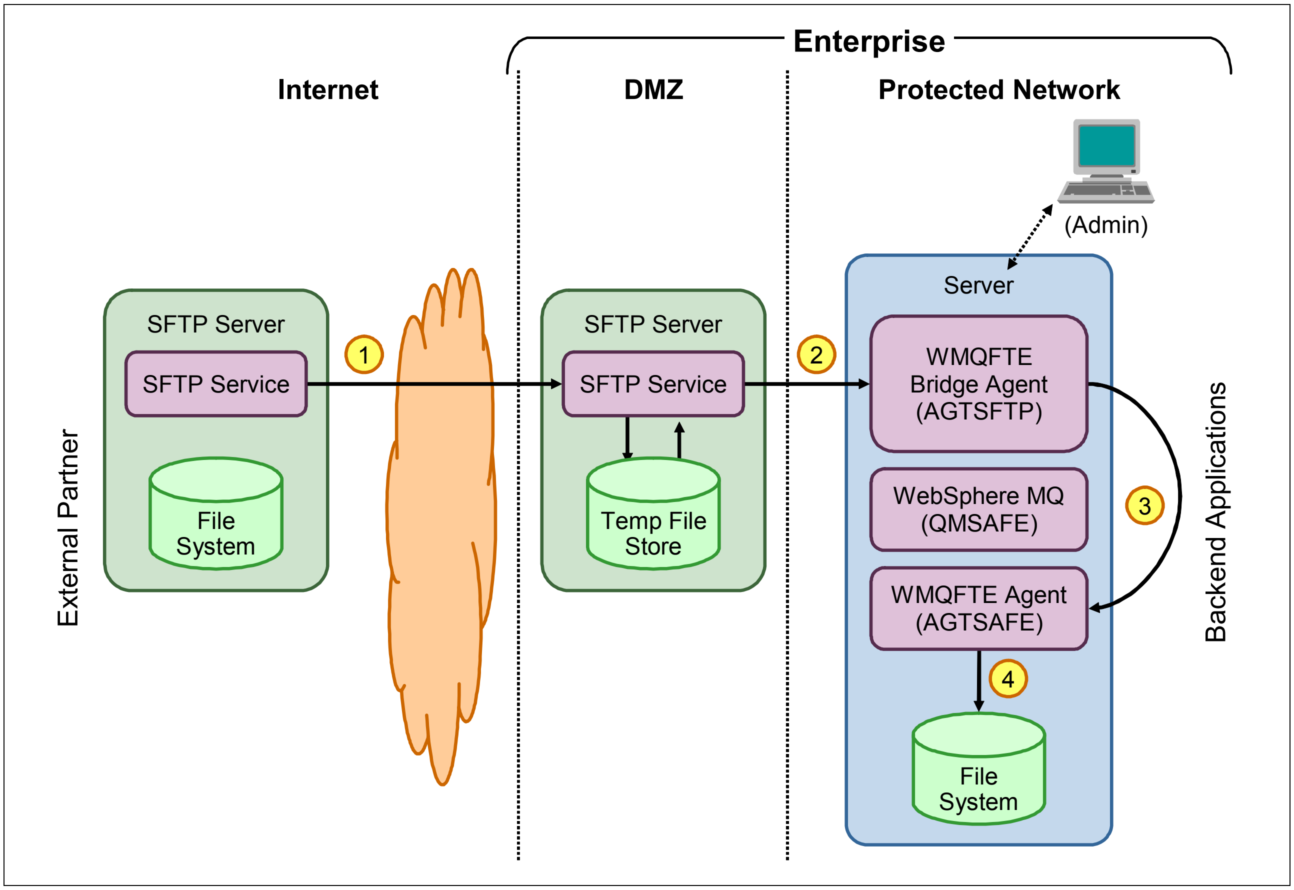


Figure 6-2 Inbound flow

The following steps are performed in the inbound scenario:

1. The external partner sends a file to the SFTP server in the DMZ. The file is stored in the temporary file system.
2. The AGTSFTP bridge agent is connected to the SFTP server in the DMZ and regularly checks the temporary file system for relevant files, based on the definition for a scheduled file transfer. When the file sent from the external partner is recognized, the AGTSFTP agent starts a file transfer and reads the file data.
3. The file data is transferred to the AGTSAFE agent.
4. The AGTSAFE agent stores the file in the file system in the internal zone.

Scenario outbound flow

Figure 6-3 shows the outbound flow.

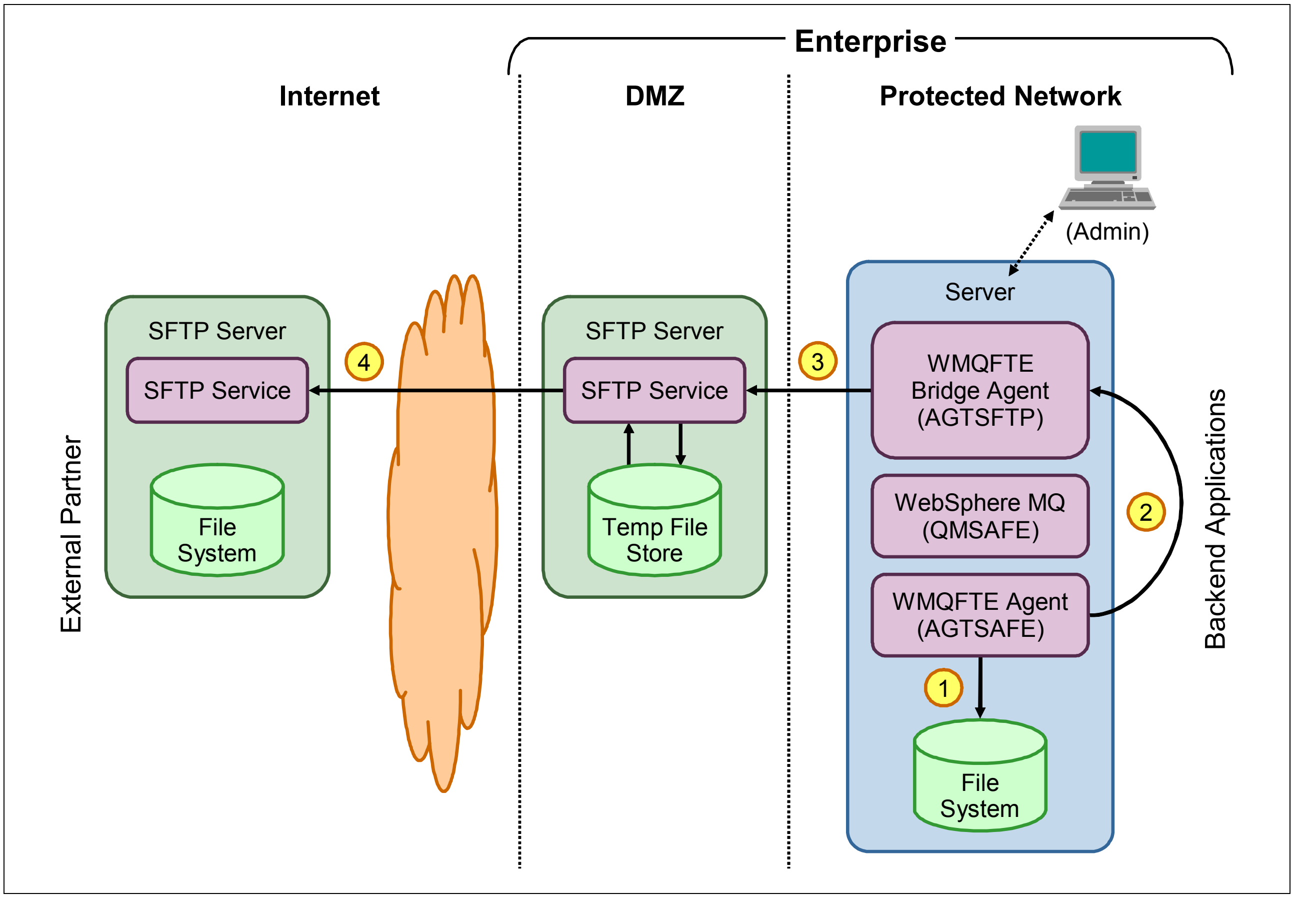


Figure 6-3 Outbound flow

The following steps are performed in the outbound scenario:

1. The AGTSAFE agent monitors the file system in the private network for outgoing files, based on the event definition for the monitor process. When a relevant file is detected, the agent starts the file transfer and reads the file data.
2. The file data is transferred to the AGTSFTP bridge agent.
3. AGTSFTP is connected to the SFTP server on the DMZ machine and transfers the file to the temporary file store in the DMZ.
4. The file is then transferred from the temporary file system in the DMZ to the SFTP server on the external partner side. This file transfer can be initiated by an SFTP client in the DMZ that pushes the file to the external partner side, or in a reverse manner the file transfer can be initiated by an SFTP client on the external partner side, which pulls the file.

6.2.3 Direct SFTP connection to external partner

This scenario describes a topology where the files from the partner are directly retrieved into the protected network. The AGTSFTP agent in the protected network is directly connected to the SFTP server on the external partner side and initiates the inbound and outbound file transfers. The transfer requests traverse the DMZ without stopping. The SFTP calls are not terminated, and no files are intermediately stored in the DMZ.

Because all file transfers in both directions are initiated and controlled by the bridge agent, this scenario shows a fully managed file transfer over the internet, leveraging the WebSphere MQ File Transfer Edition capabilities, as described in 6.1.2, “Business value” on page 86. Furthermore, because there are no requests from external partners, no inbound ports must be opened in the DMZ firewall.

Scenario inbound flow

Figure 6-4 shows the scenario flow for inbound files.

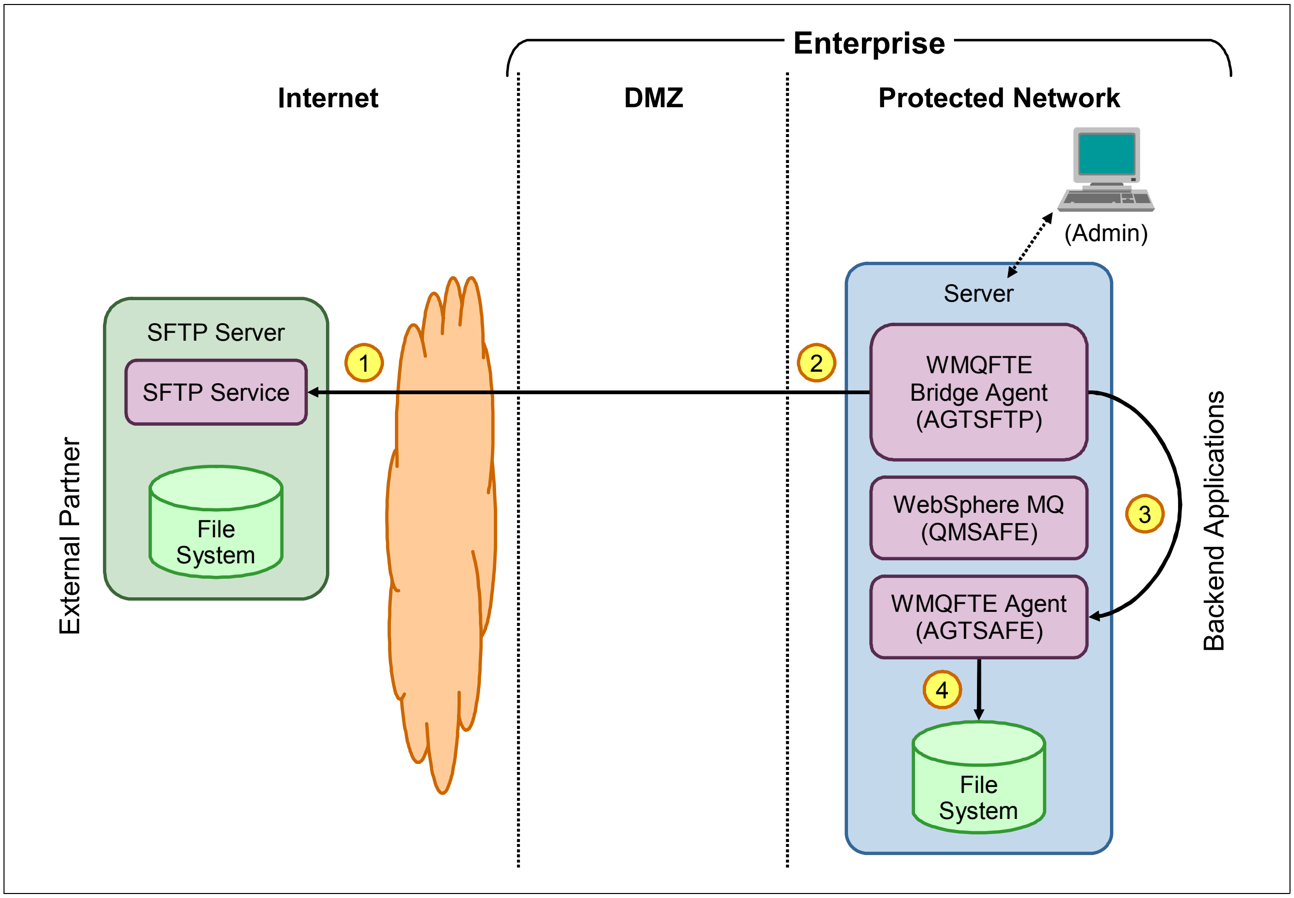


Figure 6-4 Inbound flow with direct SFTP connection to external partner

The following steps are performed in the inbound scenario:

1. The AGTSFTP agent is connected to the SFTP server on the external partner side and checks the file system for relevant files on a recurring basis that is defined in a scheduled file transfer template for this specific agent.
2. When a relevant file is recognized, the AGTSFTP agent starts the file transfer and reads the file data from the file system.
3. The AGTSFTP agent transfers the file to the AGTSAFE agent.
4. The AGTSAFE agent stores the file on the local file system.

Scenario outbound flow

Figure 6-5 shows the flow for outbound files.

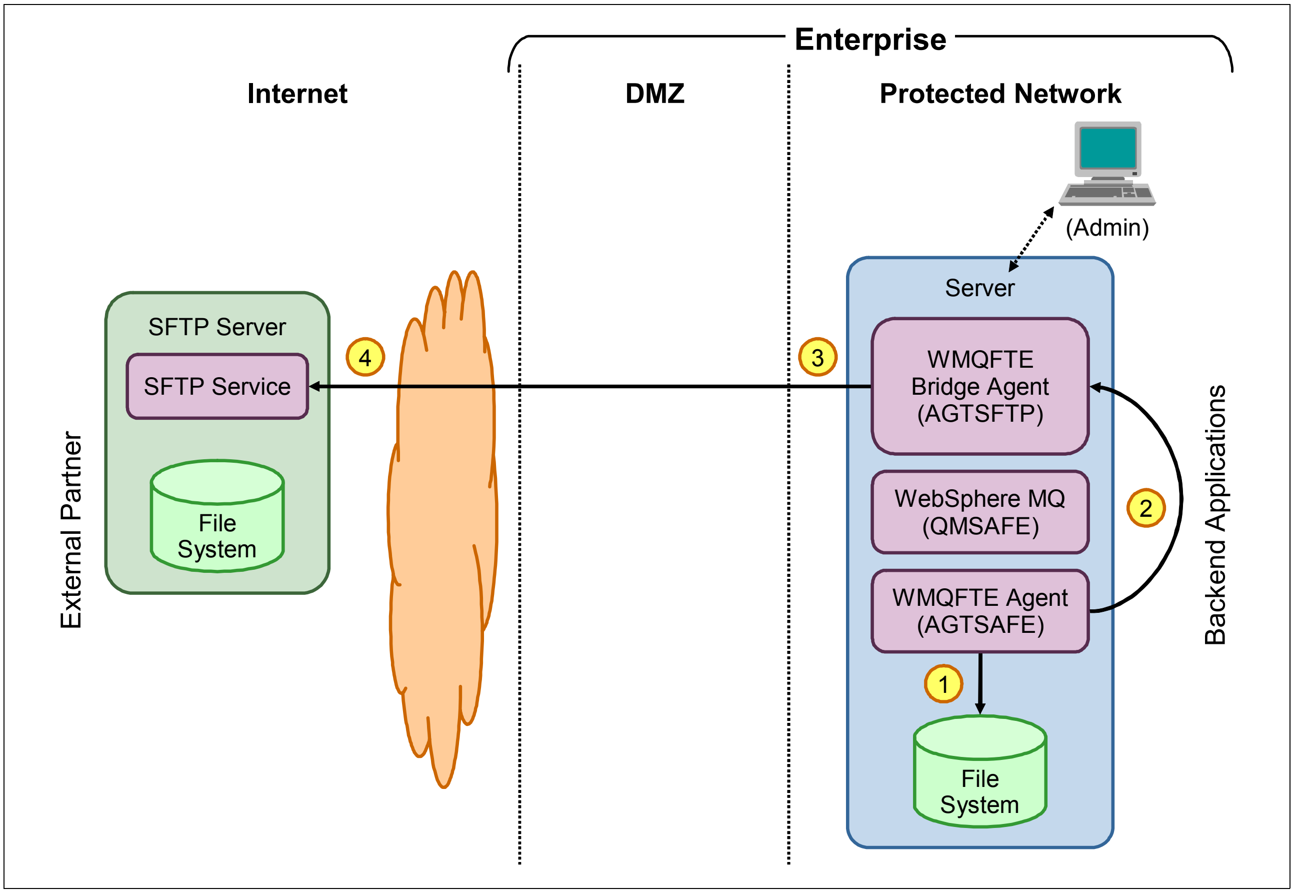


Figure 6-5 Outbound flow with direct SFTP connection to external partner

The following steps are performed in the outbound flow with a direct SFTP connection to the external partner:

1. The AGTSAFE agent monitors the file system in the internal zone for outgoing files, based on the event definition for a monitor process. When a relevant file is detected, the agent starts the file transfer and reads the file data.
2. The file data is transferred to the AGTSFTP agent.
3. The AGTSFTP agent is connected to the SFTP server on the external partner side and transfers the file over the internet to this SFTP server.
4. The file is stored in the file system on the external partner's machine.

6.2.4 Protocols

The following protocols are used in both scenarios:

- SFTP

  In the scenario in which we have a DMZ, SFTP is used for the communication between the bridge agent and the SFTP server in the DMZ, and for the communication between the SFTP server in the DMZ and the SFTP server on the external partner side.

  In the scenario with no DMZ, SFTP is used for the communication between the bridge agent and the SFTP server on the external partner side.

- WebSphere MQ

  WebSphere MQ is used for communication between the AGTSAFE agent and the AGTSFTP agent in the protected network in both scenarios.

6.2.5 Security

Because the technology and the components described in these scenarios typically are used for file transfer over the public internet, special attention must be paid to potential vulnerabilities and security requirements.

Data protection

The protocol used between the external partner and the enterprise in both scenarios is SFTP. This protocol is recognized as secure, and the data is protected while it is in transfer.

In the scenario that places a DMZ between the internet and the protected network, the file data is encrypted on the transport layer while it is transferred from the SFTP server on the DMZ machine to or from the external partner and to or from the AGTSFTP agent.

If the file data is unencrypted when it is sent from or to the partner, then it is stored unencrypted on the temporary file store. If there is a requirement to protect the file data while the files reside in the DMZ file system, then appropriate encryption mechanisms must be in place. There are different options for doing this:

- The files are encrypted prior to the transfer, using an algorithm such as pgp. This means that the files must be decrypted on the target side with the same algorithm before they can be used by an application for further processing.
- The files are encrypted when they are stored on the temporary file system. This can be done by using programs such as Encrypting File Systems (EFS) from Microsoft Windows systems or equivalent utilities for other operating systems.

User authentication and authorization on the SFTP server

The SFTP user on the external partner side must authenticate at the SFTP server in the DMZ, and reversely, the SFTP user in the DMZ must authenticate at the SFTP server on the external partner side to connect to each other.

To write and read files to and from the file systems, the users must have the appropriate access rights. The same permissions are required for the bridge agent user (the user that started the agent process).

Securing the WebSphere MQ network

WebSphere MQ File Transfer Edition uses WebSphere MQ as the underlying infrastructure. Therefore, security considerations must be made for the WebSphere MQ network and for the objects that are used by WebSphere MQ File Transfer Edition.

Although we transfer files from and to external partners in this scenario, there are no connections and no open ports in our system architecture that an external partner could use to get access to our WebSphere MQ network. Therefore, we do not address the precautionary measures for WebSphere MQ objects when external clients connect to a WebSphere MQ network in the DMZ or protected network, but we assume that the internal access and use of the WebSphere MQ objects is managed and controlled.

For more information about how to secure WebSphere MQ networks and objects see the WebSphere MQ V7 Information Center at:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp

Each WebSphere MQ File Transfer Edition agent has WebSphere MQ objects for which the user that runs the agent needs the appropriate access rights. These are the agent's system queues and, if appropriate, client channel connections. The agent's permissions should be restricted to the queues and channels that it is using.

Securing the WebSphere MQ File Transfer Edition network

WebSphere MQ File Transfer Edition has security features to protect the files and file systems from unauthorized access. These features allow organizations to control who can invoke file transfer operations, who can read and write files being transferred, and how to protect the integrity of files. This security can be achieved by:

- Managing authorities for resources specific to WebSphere MQ File Transfer Edition

  Managing authorities for WebSphere MQ File Transfer Edition resources is done through authorization queues associated with each agent. For any file transfer request, the agent process requires a level of access to its local file systems. Additionally, the user initiating the transfer and the user ID running the agent process must have authority to use certain WebSphere MQ objects.

- Managing authorities to access file systems

  The user ID running the agent process must have access to the local file system to read or write files during file transfer. This is controlled through the local operating system.

- Using sandboxes

  The access of an agent to the file system can be restricted by defining the sandboxRoot property in an agent's properties file. This property restricts the agent's access to a certain directory or a certain area of the file system, the so-called sandbox. For more information about sandboxing, see:

  http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/topic/com.ibm.wmqfte.admin.doc/sandboxes.htm.

- Using the agent's commandPath property

  The commandPath property in an agent's property file restricts the locations that an agent can run commands from. By default, the commandPath is empty, so an agent cannot call any commands. Take extreme care when this property is set because any command in one of the specified commandPath settings can be called from a remote client system that is able to send commands to the agent. For more information about this property, see:

  http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/topic/com.ibm.wmqfte.admin.doc/command_path.htm

- Configuring SSL encryption for WebSphere MQ File Transfer Edition

  The file data, when transferred between WebSphere MQ File Transfer Edition agents, can be protected by establishing Secure Sockets Layer (SSL) on the WebSphere MQ channel connections.

- Authority to publish log and status messages

  Agents issue various log, progress, and status messages to the coordination queue manager for publication. The publication of these messages can be secured using WebSphere MQ security.

6.2.6 Prerequisites

The prerequisite software for this scenario is:

- WebSphere MQ 7.0 or later
- WebSphere MQ File Transfer EditionServer 7.0.2 or later
- SFTP server (for example, OpenSSH server daemon on Linux Red Hat Enterprise)

6.2.7 Configuration prerequisites

This scenario uses the following preconfigured components for WebSphere MQ File Transfer Edition:

- WebSphere MQ queue manager QMSAFE
- WebSphere MQ File Transfer Edition agent AGTSAFE

Appendix A, “Configuration of WebSphere MQ File Transfer Edition” on page 269, provides instructions for creating these components.

For the first scenario, where the SFTP server on the external partner side transfers files to the SFTP server in the DMZ, a port must be opened in the DMZ firewall. Open a port in the DMZ firewall for SFTP communication (default for SFTP is port 22).

6.2.8 Creating and configuring the bridge agent

This section describes how to create and configure the AGTSFTP agent on the server in the protected network. We assume that the queue manager, QMSAFE, and the AGTSAFE agent are already implemented. The bridge agent requires a local agent queue manager to which to connect. In our scenario, the AGTSFTP agent is created on the same machine on which the queue manager QMSAFE resides.

1. Create the agent using the **fteCreateBridgeAgent** command.

   Open a command console and run the following command from the command line:

   ```
   fteCreateBridgeAgent -agentname AGTSFTP -agentQMgr QMSAFE -bt SFTP -bh sysa
   -btz US/Eastern -bm UNIX -bfe UTF8
   ```

   This command creates the bridge agent in bindings mode to the QMSAFE queue manager and is connected to a SFTP server on the sysa host.

   The parameters used in this command are:

   – agentname: name of the agent
   – agentQMgr: name of the agent queue manager
   – bt: protocol type (FTP or SFTP)
   – bh: host name or IP address of the SFTP server machine
   – btz: (optional) SFTP server time zone

– bm: SFTP server platform (UNIX or Windows)
– bfe: SFTP server encoding

> **For more information:** For a detailed description of the **fteCreateBridgeAgent** command, see the WebSphere MQ File Transfer Edition7.0.2 Information Center at:
>
> http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/index.jsp?topic=/com.ibm.wmqfte.home.doc/help_home_wmqfte.htmfirst

The **fteCreateBridgeAgent** command also creates three files. Two MQSC script files are created with the commands required to define and to delete the agent's system queues. It also creates a credential XML file that you must modify in a subsequent step. Information about these files is shown at the end of the command output (Example 6-1).

Example 6-1 Results of the fteCreateBridgeAgent command

```
BFGCL0069I: A file has been created containing the MQSC definitions to create
your agent. The file can be found here:
'C:\IBM\WMQFTE\config\QMSAFE\agents\AGTSFTP\AGTSFTP_create.mqsc'.
BFGCL0070I: A file has been created containing the MQSC definitions to delete
your agent. The file can be found here:
'C:\IBM\WMQFTE\config\QMSAFE\agents\AGTSFTP\AGTSFTP_delete.mqsc'.
BFGCL0277I: A credential XML file has been created.  This file must be
completed with credential details for accessing the protocol file server before
the bridge agent can be brought into service. The file can be found here:
'C:\IBM\WMQFTE\config\QMSAFE\agents\AGTSFTP\ProtocolBridgeCredentials.xml'.

BFGCL0053I: Agent configured and registered successfully.
```

Make sure that at the end of the output you see that the agent is successfully registered.

If you see a message that the agent was configured but could not be registered, this means that the coordination queue manager could not be contacted because it is not available or your configuration parameters are not correct.

The effect is that the agent can be started and transfer files, but that it is not listed by the **fteListAgents** command or in the WebSphere MQ File Transfer Edition Explorer. The status messages of this agent are also not shown in the WebSphere MQ File Transfer Edition Explorer Transfer Log view.

The WebSphere MQ reason code issued with the error provides more information about the reason for the problem. Explanations for reason codes can be found in the WebSphere MQ V7 Information Center at:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp

> **Creating the bridge agents:** The **fteCreateBridgeAgent** command creates a bridge agent for a specific SFTP server. You have to create a bridge agent for each SFTP server to which you want to connect.

2. Create the bridge agent's MQ objects on the queue manager, QMSAFE, using the script generated in the previous step. Run the AGTSFTP_create.mqsc script from the command line using the runmqsc utility:

   ```
   runmqsc QMSAFE < C:\IBM\WMQFTE\config\QMSAFE\agents\AGTSFTP\AGTSFTP_create.mqsc
   ```

   Example 6-2 Results of the AGTSFTP_create.mqsc command

   ```
   C:\>runmqsc QMSAFE < C:\IBM\WMQFTE\config\QMSAFE\agents\AGTSFTP\AGTSFTP_create
   5724-H72 (C) Copyright IBM Corp. 1994, 2009.  ALL RIGHTS RESERVED.
   Starting MQSC for queue manager QMEDGE.

   11 MQSC commands read.
   No commands have a syntax error.
   All valid MQSC commands were processed.
   ```

   Make sure that there are no errors at the bottom of the script output.

3. Configure the bridge agent credentials.

   The bridge agent user must be authenticated when the AGTSFTP agent connects to the SFTP server. The authentication of the bridge agent user at the SFTP server can be done based on user ID and password credentials or by using a public/private key pair. The **`ftecreateBridgeAgent`** command creates the `ProtocolBridgeCredentials.xml` file in which the credential mapping is for this specific agent is defined. In our scenario we use the user ID/password credentials for the authentication of our local user admin at the SFTP server.

   a. Navigate to the bridge agent's home directory:

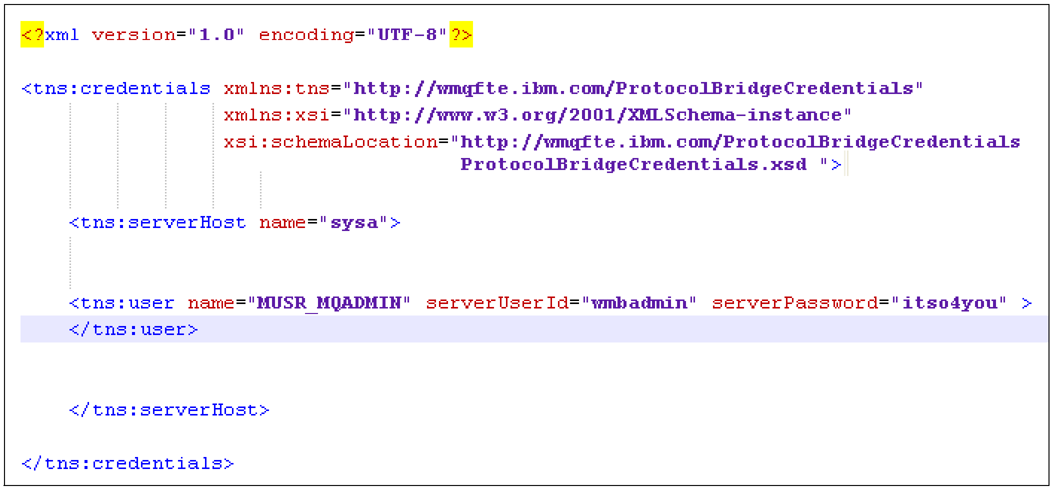
      ```
      C:\IBM\WMQFTE\config\QMSAFE\agents\AGTSFTP
      ```

   b. Edit the `ProtocolBridgeCredentials.xml` file.

   c. Insert the following credentials:

      ```
      <tns:user name="MUSR_MQADMIN"
               serverUserId="wmbadmin"
               serverPassword="itso4you" >
        </tns:user>
      ```

   Your `ProtocolBridgeCredentials.xml` file should look like Figure 6-6.



```
<?xml version="1.0" encoding="UTF-8"?>

<tns:credentials xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials"
                 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                 xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeCredentials
                                     ProtocolBridgeCredentials.xsd ">

    <tns:serverHost name="sysa">

    <tns:user name="MUSR_MQADMIN" serverUserId="wmbadmin" serverPassword="itso4you" >
    </tns:user>


    </tns:serverHost>

</tns:credentials>
```

Figure 6-6 ProtocolBridgeCredentials.xml

4. Start the bridge agent from the command line.

   Open a command console and enter the following command

   ```
   fteStartAgent AGTSFTP
   ```

   Check the agent log file to make sure that the agent started with no errors. Navigate to the `C:\IBM\WMQFTE\config\QMSAFE\agents\AGTSFTP\logs` directory and open the `output0.log` file. The file should have the entry shown in Example 6-3 at the end.

   Example 6-3 AGTSFTP log file

   ```
   Install Locations:
       com.ibm.wmqfte.product.root=C:\Program Files\IBM\WMQFTE
       com.ibm.wmqfte.product.config=C:\IBM\WMQFTE\config
   ************* End Display Current Environment *************
   [25/06/2010 17:20:29:702 EDT] 00000001 Agent        I   BFGAG0090I: This agent
   has been configured as a protocol bridge FTE agent.
   [25/06/2010 17:20:29:702 EDT] 00000001 Agent        W   BFGAG0125W: The
   maximum size to which the java heap can grow is '550'MB, which is the default
   value. This value may be too low dependent on the agent's work load.
   [25/06/2010 17:20:29:702 EDT] 00000001 AgentRuntime  I   BFGAG0058I: The agent
   has successfully initialized.
   [25/06/2010 17:20:30:936 EDT] 00000001 AgentRuntime  I   BFGAG0059I: The agent
   has been successfully started.
   ```

   Another option for checking whether the agent is running is to ping the agent using the **ftePingAgent** command (Example 6-4).

   Example 6-4 Using the ftePingAgent command

   ```
   ftepingagent AGTSFTP
   5655-U80, 5724-R10 Copyright IBM Corp.  2008, 2009.  ALL RIGHTS RESERVED
   BFGCL0212I: Issuing ping request to agent AGTSFTP
   BFGCL0213I: agent AGTSFTP responded to ping in 0.422 seconds.
   ```

6.2.9 Defining a scheduled file transfer

All file transfers in and out of the protected network are initiated by the bridge agent. To automate the transfer of the files from the SFTP server, a mechanism must be implemented to check the file system on the remote SFTP server machine and start the file transfer when relevant files are found. For a bridge agent, this mechanism is implemented using a scheduled file transfer. A scheduled file transfer can be defined in the WebSphere MQ Explorer or by creating a transfer template using the **fteCreateTemplate** command. In this scenario we create the template from the command line.

To create the template, open a command console and run the following command:

```
fteCreateTemplate -tn From_Partner -sa AGTSFTP -da AGTSAFE -ss 07:05 -oi minutes
-of 10 -de overwrite -sd delete -dd D:\partners\partner1\In To_Enterprise/po*.xml
```

The parameters used in this command are:

- tn: name of the template
- sa: name of the source agent
- da: name of the destination agent
- ss: start time
- oi: occurrence interval
- of: occurrence frequency

- de: file behavior at destination
- sd: source file disposition
- dd: destination directory

For a detailed description of the **fteCreateTemplate** command see the WebSphere MQ File Transfer Edition7.0.2 Information Center at:

http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/index.jsp?topic=/com.ibm.wmqfte.home.doc/help_home_wmqfte.htmfirst

The command creates a transfer template with the name `From_Partner`, which initiates the AGTSFTP agent to check the `To_Enterprise` folder every 10 minutes. It is important to note that we use a wildcard character for the name of the files for which we are looking. The relevant file name starts with `po`, and the asterisk (*) represents zero or more following wildcard characters in the file name. For example, matching file names are poForYOU.xml and po_abc123.xml.

> **Determining the match pattern:** When we test the inbound scenario later in this chapter (see 6.3.1, "Testing inbound scenario with direct SFTP connection to external partner" on page 103), we send purchase orders from the external partner to the enterprise. These files have file names starting with `po`, which is the reason why we define the scheduled file transfer with this file name match pattern.

If one or more relevant files are found, the AGTSFTP agent starts the file transfer and transfers the files to the AGTSAFE agent, which stores the files in `D:\partners\partner1\In`. The source files are deleted, and files in the target directory are overwritten if they already exist.

Check in the WebSphere MQ Explorer Transfer Template view to ensure that the template has been created successfully. The template can be modified and duplicated for further re-use.

There is one more step to do. A scheduled transfer created using the **fteCreateTemplate** command is not active automatically. We must submit the template to activate the scheduled file transfer. Figure 6-7 shows how to submit the template using the WebSphere MQ Explorer.

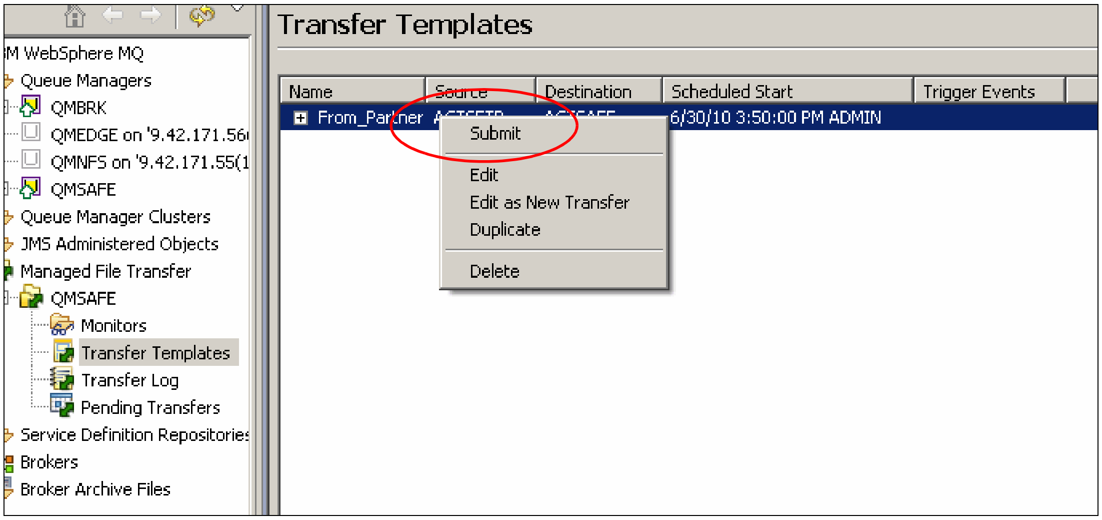


Figure 6-7 Submit the transfer template

You can check whether the scheduled transfer is active. Open a command window and issue the **fteListScheduledTransfers** command (Example 6-5).

Example 6-5 Verify that the scheduled transfer is active

```
fteListScheduledTransfers
5655-U80, 5724-R10 Copyright IBM Corp.  2008, 2009.  ALL RIGHTS RESERVED

Schedule Identifier:        5
Source Agent Name:          AGTSFTP
Source File Name:           To_Enterprise/po*.xml
Conversion Type:            binary
Destination File Name:      D:\partners\partner1\In
Destination Agent Name:     AGTSAFE
Schedule Start Time:        2010-07-12T15:50-0400
Next Transfer:              2010-07-12T15:50-0400
Schedule Time Base:         admin
Repeat Interval:            minutes
Repeat Frequency:           10
```

In the command output, the **fteListScheduledTransfers** command shows an integer as the schedule identifier, rather than the name that we defined in the fteCreateTemplate statement. This schedule identifier can be used to delete the scheduled transfer, using the **fteDeleteScheduledTransfer** command.

6.2.10 Defining a resource monitor

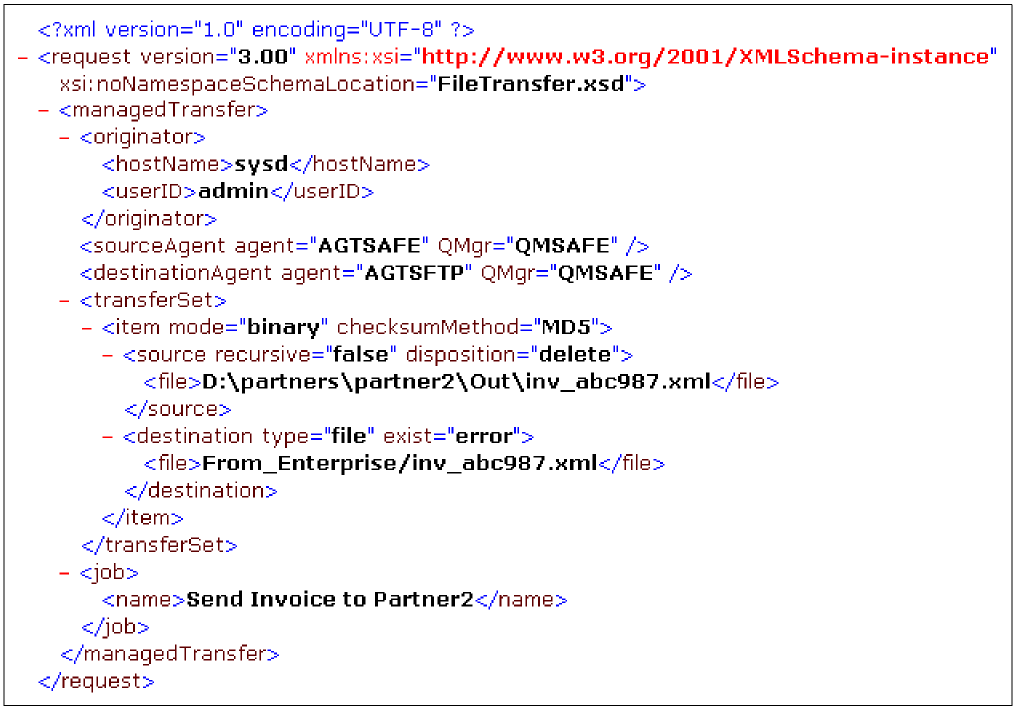
With a resource monitor definition, an agent can monitor a specific file system or folder and initiate a file transfer when a certain event occurs (for example, a file with a given match pattern is recognized). For the outbound file transfers, a resource monitor is defined for the AGTSAFE agent to poll the local file system and start the file transfer when a relevant file is found. A resource monitor can be created either using the WebSphere MQ Explorer or the command-line interface.

The following steps describe how to define a resource monitor using the **fteCreateMonitor** command on the command line:

1. Define a task definition file.

   The task definition file contains the parameters for the file transfer. The file can be used by the resource monitor, rather than defining the parameters in the command line. The task definition file must comply with the schema `Filetransfer.xsd` and should have the `<managedTransfers>` element as the root element. You can find the schema `Filetransfer.xsd` in the WebSphere MQ File Transfer Edition `<`*`install-dir`*`>\tools\samples\schema` folder.

For this scenario, the task definition file has the attributes shown in Figure 6-8 defined. This file is stored as `C:\tmp\transfer_def_file.xml`.



```
  <?xml version="1.0" encoding="UTF-8" ?>
– <request version="3.00" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  – <managedTransfer>
    – <originator>
        <hostName>sysd</hostName>
        <userID>admin</userID>
      </originator>
      <sourceAgent agent="AGTSAFE" QMgr="QMSAFE" />
      <destinationAgent agent="AGTSFTP" QMgr="QMSAFE" />
    – <transferSet>
      – <item mode="binary" checksumMethod="MD5">
        – <source recursive="false" disposition="delete">
            <file>D:\partners\partner2\Out\inv_abc987.xml</file>
          </source>
        – <destination type="file" exist="error">
            <file>From_Enterprise/inv_abc987.xml</file>
          </destination>
        </item>
      </transferSet>
    – <job>
        <name>Send Invoice to Partner2</name>
      </job>
    </managedTransfer>
  </request>
```

Figure 6-8 Task definition file

> **UserID attribute:** The UserID attribute must contain the user ID of the MQMD header field. This is the user that you used to start the WebSphere MQ service on that machine.

2. Create the resource monitor for the AGTSAFE agent.

   Open a command console and run the following command from the command line:

   ```
   fteCreateMonitor -ma AGTSAFE -mm QMSAFE -md D:\partners\partner2\Out -mn
   Partner2_Out -mt C:\tmp\transfer_def_file.xml -pi 30 -pu seconds -tr
   match,inv*.xml
   ```

   The parameters used in this command are:

   – ma: name of the agent that executes the monitoring task
   – mm: agent queue manager of the monitoring agent
   – md: directory that is to be monitored
   – mn: monitor name
   – mt: task definition file that is executed when the file transfer is initiated
   – pi: interval period
   – pu: unit for the interval period
   – tr: match criteria

   For a detailed description of the **`fteCreateMonitor`** command, see the WebSphere MQ File Transfer Edition7.0.2 Information Center at:

   `http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/index.jsp?topic=/com.ibm.wmqfte.home.doc/help_home_wmqfte.htmfirst`

In our monitor definition, the AGTSAFE agent polls the `D:\partners\partner2\Out` directory every 30 seconds for xml files with a file name that starts with `inv`.

When a relevant file is found, a file transfer is initiated based on the parameters defined in the `C:\tmp\transfer_def_file.xml` transfer definition file (which we created in the previous step).

> **Determining the match pattern:** When we test the outbound scenario later in this chapter (see 6.3.2, “Testing outbound scenario with direct SFTP connection to external partner” on page 104), we send an invoice from the enterprise to the external partner. These files have file names starting with `inv`, which is why we define the scheduled file transfer with this file name match pattern.

You can check whether the resource monitor is up and running by issuing the **`fteListMonitors`** command (Example 6-6).

Example 6-6 fteListMonitors command

```
fteListMonitors
5655-U80, 5724-R10 Copyright IBM Corp.  2008, 2009.  ALL RIGHTS RESERVED
Agent Name:     Monitor Name:
AGTSAFE         PARTNER2_OUT
```

6.3 Testing the scenarios

We created and configured the SFTP bridge agent AGTSFTP, and we have defined a scheduled file transfer for this agent for the retrieval of files from the external partner. Furthermore, we defined a resource monitor for the AGTSAFE agent to check for outgoing files to the external partner.

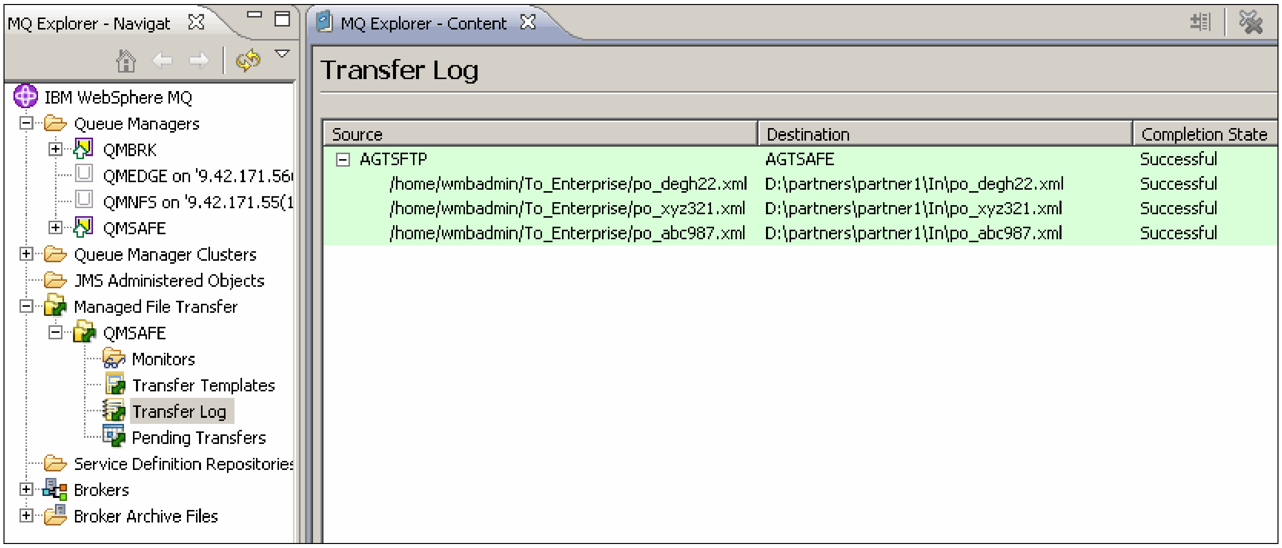
We now test our configuration for inbound and outbound file transfers, based on the scenario in which we pass directly through the DMZ between the internet and the protected network (6.2.3, “Direct SFTP connection to external partner” on page 91).

6.3.1 Testing inbound scenario with direct SFTP connection to external partner

In our scenario we copy the three test files (`po_abc987.xml`, `po_degh22.xml`, and `po_xyz321.xml`) into the `To_Enterprise` folder on the SFTP server on the external partner side.

The AGTSFTP agent is connected to the SFTP server on the external partner side and is scheduled to check the `To_Enterprise` folder every 10 minutes for xml files with the pattern `po` at the beginning of the file name. All the files that we have copied there match this pattern, so the expected result is that after no longer than 10 minutes all the files will be transferred and subsequently deleted from the SFTP server file system.

Check the transfer status in the WebSphere MQ Explorer using the Transfer Log view. You should see that all three files have transferred successfully (Figure 6-9).



Transfer Log

Source	Destination	Completion State
⊟ AGTSFTP	AGTSAFE	Successful
/home/wmbadmin/To_Enterprise/po_degh22.xml	D:\partners\partner1\In\po_degh22.xml	Successful
/home/wmbadmin/To_Enterprise/po_xyz321.xml	D:\partners\partner1\In\po_xyz321.xml	Successful
/home/wmbadmin/To_Enterprise/po_abc987.xml	D:\partners\partner1\In\po_abc987.xml	Successful

Figure 6-9 Transfer log

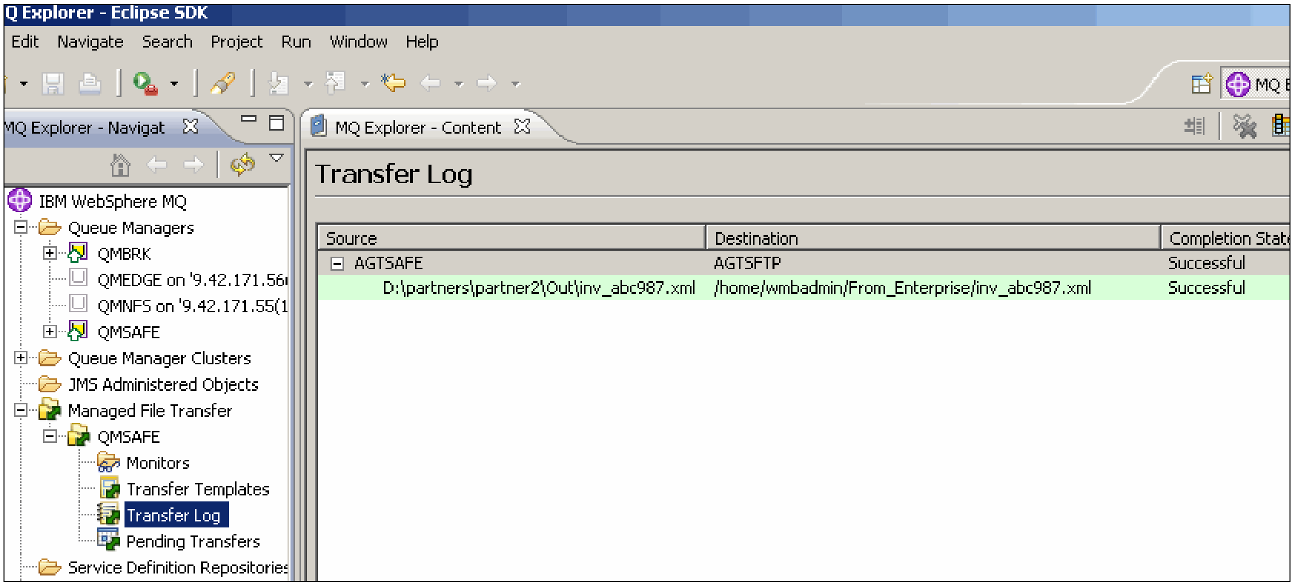
As you can see in the Transfer Log view, all three files are transferred within one single transfer request. A WebSphere MQ File Transfer Edition agent can transfer multiple files and entire file system directories in a single file transfer request.

6.3.2 Testing outbound scenario with direct SFTP connection to external partner

For this test, we move or copy a test file named inv_`abc987.xml` into the `D:\partners\partner2\Out` folder on the local file store in the protected network. The file name corresponds with the match criteria that we defined in our `fteCreateMonitor` statement.

The AGTSAFE agent polls the `D:\partners\partner2\Out` folder. The expected result is that after no longer than 30 seconds you will see the file disappear from the source directory. This means that the file was recognized by the AGTSAFE agent and successfully transferred to the AGTSFTP agent.

If you want to check the status of the file transfer, open the Transfer Log view in your WebSphere MQ Explorer. You should see a log entry that the file was successfully transferred from the `D:\partners\partner2\Out` folder in the file system in the protected network to the `From_Enterprise` folder in the file system of the external partner side (Figure 6-10).



Transfer Log

Source	Destination	Completion Stat
⊟ AGTSAFE	AGTSFTP	Successful
D:\partners\partner2\Out\inv_abc987.xml	/home/wmbadmin/From_Enterprise/inv_abc987.xml	Successful

Figure 6-10 Transfer log for the outbound test

6.4 Troubleshooting tips for WebSphere MQ File Transfer Edition

This section provides information about troubleshooting WebSphere MQ and WebSphere MQ File Transfer Edition and how to diagnose errors that could occur in the scenarios that we describe in this section.

Detailed information about WebSphere MQ File Transfer Edition diagnostic messages can be found in the information center at:

http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/index.jsp?topic=/com.ibm.wmqfte.messages.doc/messages_main.htm

Detailed information about WebSphere MQ error codes can be found in the information center at:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp

6.4.1 Checking the system requirements

It is important for the installation, configuration, and operation of all the WebSphere MQ and WebSphere MQ File Transfer Edition objects that your systems are compliant with the system requirements for the software. The system requirements can be found at:

http://www-01.ibm.com/software/integration/wmq/filetransfer/requirements

If you are in doubt, contact IBM and ask for help.

6.4.2 Log files

When errors occur or file transfers do not start, a good first place to look is the log files:

- Agent log file (`output0.log`)

  The agent log file is created in the agent's log directory:

  `<configuration_directory>\<coordination_qmgr_name>\agents\<agent_name>\logs`

  For example, for the AGTSAFE agent the log directory is:

  `C:\IBM\WMQFTE\config\QMSAFE\agents\AGTSAFE\logs`

  The log contains records of the agent's events.

- `agent.lck` file

  This file is created in the agent's config directory:

  `<configuration_directory>\<coordination_qmgr_name>\agents\<agent_name>\logs`

  It contains the agent's process ID (PID), if it is running.

- FFDC/ABEND files

  These files are created in the agent's log directory at:

  `<configuration_directory>\<coordination_qmgr_name>\agents\<agent_name>\logs`

  They are often created as a result of an unexpected error. Analyzing the information in these files requires assistance from IBM support or IBM service.

- Queue manager log files

  Log files are created for each queue manager in the WebSphere MQ `<installation_directory>\Qmgrs\<queue_manager_name>\errors`.

6.4.3 Gathering diagnostics by enabling trace files

Tracing can be enabled for each agent by executing the following commands:

- **`fteSetAgentTraceLevel -traceLevel all`**

  This sets the trace level for an already running agent.

- **`fteStartAgent -trace =all`**

  This starts the agent with trace enabled.

The trace output from both commands goes into the agent's log directory `<configuration_directory>\<coordination_qmgr_name>\agents\<agent_name>\logs`.

The following trace levels can be set:

- off (This is the default.)
- flow
- moderate
- all

Be careful when enabling tracing for agents because that can have a significant impact on performance and produce a huge amount of data.

6.4.4 No agent listed in WebSphere MQ File Transfer Edition Explorer or by fteListAgents command

If you have created an agent using the **fteCreateAgent** or **fteCreateBridgeAgent** command, but your agent is not listed in WebSphere MQ Explorer or is not shown when you execute the **fteListAgents** command, the problem could be the result of any one of a number of causes. The flow chart in Figure 6-11 can help you identify the cause.

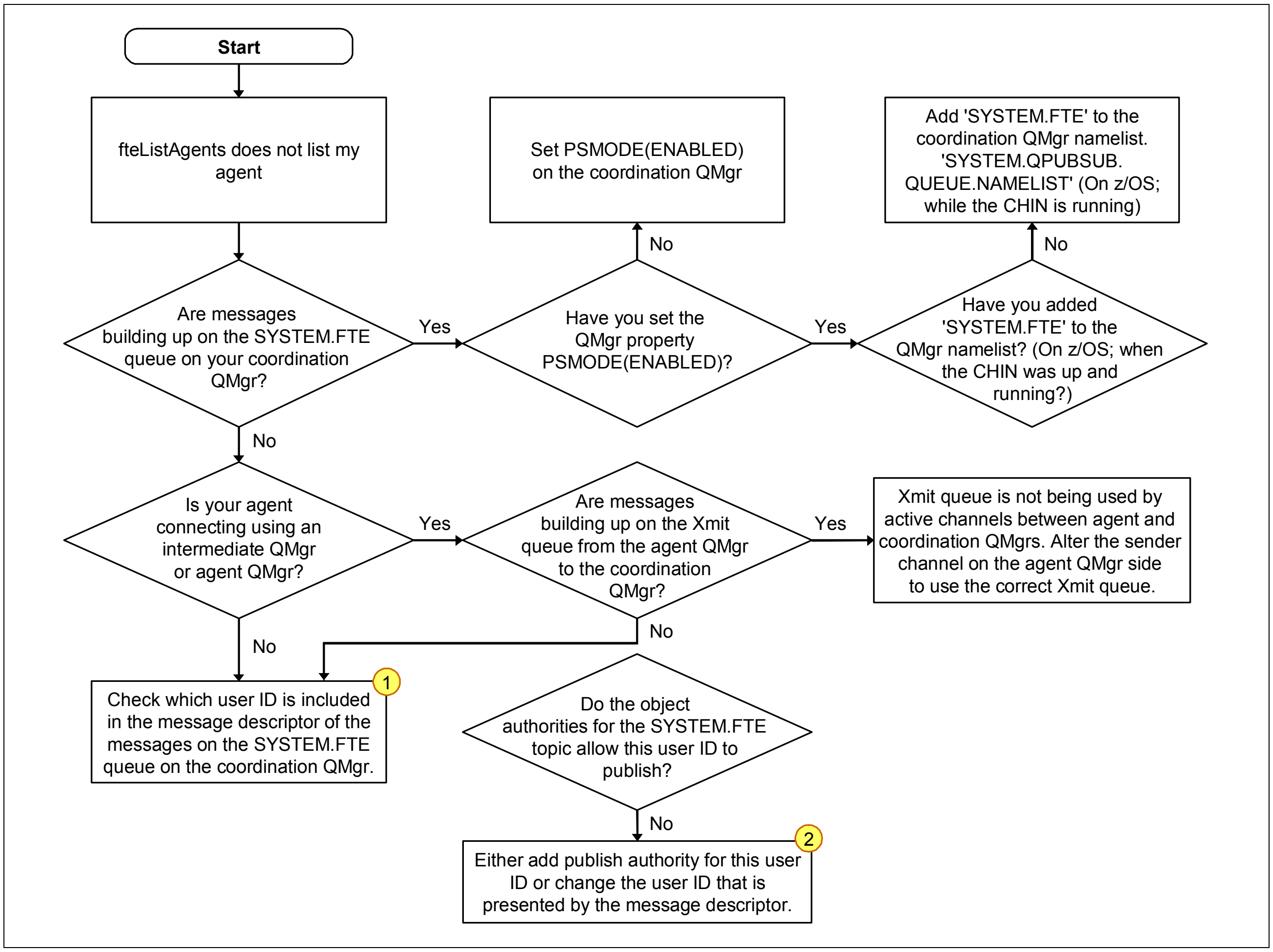

Start
fteListAgents does not list my agent
Are messages building up on the SYSTEM.FTE queue on your coordination QMgr?
Yes
Have you set the QMgr property PSMODE(ENABLED)?
No
Set PSMODE(ENABLED) on the coordination QMgr
Yes
Have you added 'SYSTEM.FTE' to the QMgr namelist? (On z/OS; when the CHIN was up and running?)
No
Add 'SYSTEM.FTE' to the coordination QMgr namelist. 'SYSTEM.QPUBSUB. QUEUE.NAMELIST' (On z/OS; while the CHIN is running)
No
Is your agent connecting using an intermediate QMgr or agent QMgr?
Yes
Are messages building up on the Xmit queue from the agent QMgr to the coordination QMgr?
Yes
Xmit queue is not being used by active channels between agent and coordination QMgrs. Alter the sender channel on the agent QMgr side to use the correct Xmit queue.
No
No
1
Check which user ID is included in the message descriptor of the messages on the SYSTEM.FTE queue on the coordination QMgr.
Do the object authorities for the SYSTEM.FTE topic allow this user ID to publish?
No
2
Either add publish authority for this user ID or change the user ID that is presented by the message descriptor.

Figure 6-11 Flowchart for debugging an agent problem

Visit the following web site for further information about how to solve the problem:

http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/index.jsp?topic=/com.ibm.wmqfte.admin.doc/list_agents_pd.htm

6.4.5 File transfer does not start

There can be various reasons for a file transfer request failing to start. The following steps can help you find the problem:

1. Determine whether the transfer requests are getting to the agent.
   a. Stop the agent.
   b. Send a new transfer request.

    c. Check to see whether the request arrives on the agent’s command queue (the SYSTEM.FTE.COMMAND.*<agent_name>* at the agent queue manager).

2. Make sure that all the relevant channels between the queue managers (cluster-sender, cluster-receiver) are up and running.
3. Check whether the source agent is running and has started the transfer.

    a. Ping the agent using the **`ftePingAgent`** command.

    b. Check the agent’s command queue. It should be empty.

    c. Use the **`fteListAgents`** and **`fteShowAgentDetails`** commands to show information about current transfers (use the -v option).

4. Determine whether the transfer has taken place but is not shown in the WebSphere MQ File Transfer Edition Explorer Log view. Make sure that the user in the MQMD user field is authorized to send status messages to the coordination queue manager.

B2B-enabled managed file transfer

In this chapter we outline a common scenario for business-to-business (B2B) enabled multi-enterprise file transfers. B2B-enabled file transfer is the combination of using B2B messaging protocols, such as EDIINT AS1, AS2, and AS3, and partner profile management at the edge of the network and a file transfer backbone inside the protected network to move files securely between external partners and back-end systems.

Additional material: Additional material is provided in support of the activities in this chapter. If you have access to the components needed to build this solution and would like to try this on your own, the additional material can be downloaded to assist you with your configuration and with testing the setup. For more information about downloading the additional material and its contents, see Appendix C, “Additional material” on page 317. The files for this chapter can be found in the `Common_Files` and the `B2BScenario_Files` directories.

7.1 Scenario overview

Many organizations devote valuable IT resources to building and maintaining systems in-house for moving files between applications. Most of these solutions are based on File Transfer Protocol (FTP) because of its simplicity and free availability. Whereas FTP offers a basic mechanism for file sharing when several applications access an occasionally updated, centrally managed source file, some enterprises are seeking alternatives for files that are moved between applications as part of business transactions. As volumes of transfers rapidly grow, and with increased consequences for errors in business data when it is incorrectly transferred, a reliable, flexible, cost-effective solution for managed file transfer is increasingly critical for organizations of all sizes. Additionally, these organizations need to enable transfers across boundaries with their trading partners and need to support a wide range of B2B and non-B2B protocols with the ability to ensure data security and partner identity while the files traverse the internet.

This scenario can support the use of any transport and B2B protocol that the DataPower B2B Appliance XB60 (referred to as the XB60 from here on) has available. However, for the purpose of the scenario flows demonstrated in this chapter, we utilize a predominate B2B messaging protocol, AS2. This allows us to demonstrate:

- How the XB60 can use profile management to verify and validate trading partners
- How B2B messaging can be used to protect the payload data that is transferred between you and your external partners
- How the XB60 can provide non-repudiation of origin and receipt for the public side of the connection (*assured delivery*)
- How the XB60 integrates with WebSphere MQ File Transfer Edition to facilitate an intra-enterprise file transfer to and from any location inside your enterprise

7.1.1 Appropriate use

This scenario demonstrates AS2 communications between the external partner and the B2B gateway representing the internal partner. However, this scenario can be varied to use any of the B2B messaging protocols supported on the XB60. The XB60 V3.8.1.2 firmware supports AS1, AS2, AS3, and ebMSv2 B2B messaging protocols. To vary the scenario to utilize one or more of these protocols you simply need to configure front-side protocol handlers for each protocol, associate them with the B2BFTE_HUB gateway, and configure partner destinations that utilize the B2B protocol.

There are many ways to vary the WebSphere MQ File Transfer Edition deployment to meet specific needs, using more sophisticated WebSphere MQ File Transfer Edition topologies. These topologies are described in detail in *Getting Started with WebSphere MQ File Transfer Edition V7*, SG24-7760.

You might also choose to make the solution highly available. The XB60 provides the ability to provide high availability in an active/standby appliance deployment model. Detailed information about XB60 high availability can be found in the WebSphere DataPower user documentation, which can be downloaded from the IBM information center at:

http://publib.boulder.ibm.com/infocenter/wsdatap/v3r8m1/index.jsp

MQ File Transfer Edition can also be made highly available. Detailed information about WebSphere MQ File Transfer Edition high availability can be found in *Getting Started with WebSphere MQ File Transfer Edition V7*, SG24-7760.

7.1.2 Business value

There is significant business value in combining the WebSphere DataPower B2B Appliance XB60 and WebSphere MQ File Transfer Edition to enable reliable and auditable internal file transfers and securing external file transfers between organizations by providing B2B governance and security at the edge of the network.

The following list describes the combined benefits that you can expect from this type of deployment scenario:

- The XB60 provides exceptional data security and certificate management that is built into the appliance. It provides robust authentication, authorization, and auditing (AAA) capabilities and built-in integration to external repositories.
- Access to files is controlled by file system permissions. File transfers can be protected using SSL encryption and authentication.
- The XB60 provides simplified deployment and ongoing management.
- The XB60 reduces the need for in-house skills that are typically needed to deploy and manage B2B-enabled file transfer solutions.
- File transfers can be set up to occur at specified times or dates, or repeated at specified intervals. File transfers can also be triggered by a range of system events, such as new files or updated files.
- The B2B appliance provides robust logging and support for saving logs to a large variety of log targets utilizing a broad range of log formats. WebSphere MQ File Transfer Edition provides full logging of transfers at both the source and destination systems for internal transfers.
- The B2B appliance is a hardened drop-in network device that is suitable for DMZ deployments, uses dedicated, tightly optimized hardware and firmware, and has no software to install.
- The B2B appliance provides the ability for external partners to view the state of their own transactions, and they can be authorized to have the ability to manually resend transactions to themselves.
- The B2B appliance supports a wide range of protocols to allow flexibility for connecting to external partners with varying skill levels and connectivity capabilities.
- Dedicated B2B appliances have been shown to reduce deployment and operational costs by as much as 50%. Their use dramatically decreases the testing time and amount of development required to upgrade your environment. Most policies are configuration-driven as opposed to development-driven.

7.2 Scenario details

This scenario has one inbound flow and one outbound flow. Each flow uses an internal partner named B2BFTE and an external partner named PARTNER. Trading partners are either internal partners or external partners that are associated with the B2B Gateway service (named B2BFTE_HUB) in the XB60. Each flow also utilizes WebSphere MQ File Transfer Edition agents named AGTNFS and AGTSAFE to move EDI X12 files. This scenario demonstrates both inbound and outbound delivery and reception of AS2 messages, AS2 packaging and unpackaging, and the routing of the payload (file) to and from WebSphere MQ File Transfer Edition.

7.2.1 Solution components

This section describes the configuration objects associated with each product in the solution.

XB60 configuration objects

This section describes each configuration object used in the XB60 to support this scenario. The XB60 provides the ability to configure the appliance utilizing three options:

- A web-based graphical user interface
- A command-line interface
- An Extensible Markup Language (XML) SOAP management interface

For the purposes of this scenario we use the web-based graphical user interface.

External partner profile

External partner profiles represent the companies that do business with you, the B2B hub owner. External partners must complete a configuration process on their B2B hub to configure an external representation of your profile. After you are connected, external partners can exchange electronic business documents with your hub. In this scenario the external partner's company is represented by a separate application domain named PARTNER configured in the XB60. The external partner profile in this scenario is represented in the B2B gateway (B2BFTE_HUB) by an external profile object named PARTNER.

Internal partner profile

The internal partner profile within your hub typically is associated with your company or a department inside your company. Your company typically is responsible for the purchase and construction of the hub product, including definition of the electronic business processes transacted between your company and your external partners. The internal partner's company is represented by a separate application domain named B2BFTE. The internal partner profile in this scenario is represented in your B2B gateway (B2BFTE_HUB) by a profile object named B2BFTE.

B2B Gateway service

The B2B Gateway service is a configuration object that is responsible for processing and routing B2B data. All data that passes through the B2B Gateway service is tied to profile management, meaning that partner information is either extracted from the payload or it is set using XSLT. This service automatically extracts partner IDs from EDI X12 and EDIFACT document types based on the standards that govern each B2B document type. Partner IDs are extracted from XML utilizing XPaths configured in the B2B Gateway service. Binary files cannot be parsed for partner IDs and must use the document routing preprocessor in the B2B Gateway service to set the IDs. Additionally, all data that passes through this service is persisted to a hard drive for non-repudiation, and all metadata associated with the transaction is persisted in a data store and is visible to the user in the B2B transaction viewer. The B2B Gateway service in this scenario is named B2BFTE_HUB.

Multi-protocol gateway service

The multi-protocol gateway service is a configuration object that is responsible for processing and routing any data type. It is essentially a secure router and firewall type of service. Unlike the B2B Gateway service, this service does not persist any data and is not tied to profile management or to the B2B viewer. The multi-protocol gateway service in this scenario is named is named MQFTE_INTEGRATION. It is used as a post process attached to the B2B Gateway service for integration with WebSphere MQ File Transfer Edition.

B2B transaction viewer

The B2B transaction viewer is used to view all transactions that pass through a B2B Gateway service. This service can be used to manually resend transactions from the XB60 to external partners. You can optionally allow external partners access to view the state of their transactions by using the appliance's access control and role-based management capabilities.

MQ File Transfer Edition configuration objects

This section describes each configuration object used in MQ File Transfer Edition to support this scenario.

WebSphere MQ queue managers

The scenario uses two WebSphere MQ queue managers:

- QMNFS
- QMSAFE

QMNFS runs on the server with the Network File System (NFS) file share and receives the XML message from the DataPower XB60 to initiate a file transfer. QMNFS is the command and agent queue manager for the AGTNFS agent. The queues that AGTNFS uses to perform a file transfer are managed and found in QMNFS.

QMSAFE runs on the Enterprise Service Bus (ESB) server and represents the queue manager for the back-end application that receives or sends a file transfer. QMSAFE is the coordination queue manager for the WebSphere MQ File Transfer Edition environment. QMSAFE also acts as the command and agent queue manager for the AGTSAFE agent. The queues that AGTSAFE uses for the file transfer are managed by QMSAFE. Additionally, interested parties can subscribe to topics associated with the file transfer on QMSAFE through topic strings. Messages published through QMSAFE are for audit and transfer monitoring purposes.

Server agents

AGTSAFE and AGTNFS are both server agents. Server agents make a bindings connection to their queue manager. The bindings connection allows the agents to communicate with their managers through cross-memory calls.

The server agent is supplied with WebSphere MQ File Transfer Edition Server licensing.

WebSphere MQ Explorer

You can administer WebSphere MQ File Transfer Edition with the WebSphere MQ Explorer workbench, using the WebSphere MQ File Transfer Edition GUI plug-in. This plug-in is part of the IBM WebSphere MQ File Transfer Edition Remote Tools and Documentation product.

WebSphere MQ Explorer is available for Windows and Linux platforms, is supplied with WebSphere MQ, and is in stand-alone form with WebSphere MQ MS0T SupportPac.

7.2.2 Scenario flow inbound

Figure 7-1 depicts the inbound data flow for this scenario.

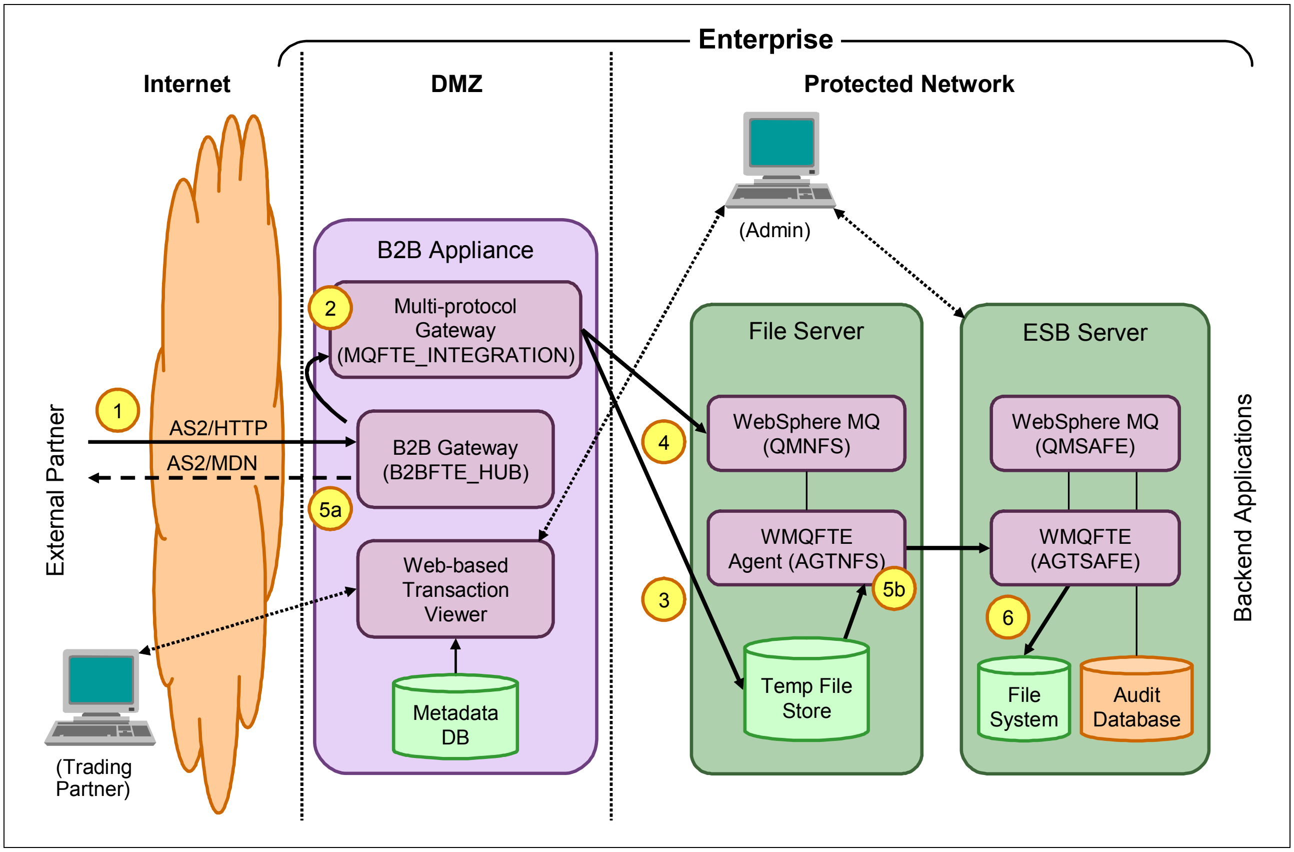


Figure 7-1 Inbound data flow

The flow shown in Figure 7-1 is:

1. The trading partner sends an AS2 message with an EDI X12 payload into the B2BFTE_HUB gateway. The B2B Gateway service uses profile management to verify and validate the partner, ensures that the message adheres to the AS2 standard, verifies that data security is appropriately applied, and removes the AS2 protocol packaging.
2. The EDI X12 payload is routed into a multi-protocol gateway service, which uses a processing policy to facilitate integration to WebSphere MQ File Transfer Edition.
3. The multi-protocol gateway policy sends the payload to an NFS mount point that is shared between the XB60 and the WebSphere MQ File Transfer Edition.
4. The multi-protocol gateway processing policy sends an XML transfer request message to the WebSphere MQ File Transfer Edition command queue. This message triggers the transfer.
5. Steps 5a and 5b happen in parallel:
   a. An AS2 Message Disposition Notification is sent to the external partner to inform them of the successful file transfer.
   b. AGTNFS reads the file from the temporary storage on the Linux file server and sends the file to the AGTSAFE.
6. AGTSAFE writes the file to a file system monitored by a back-end application.

7.2.3 Outbound data flow scenario

Figure 7-2 depicts the outbound data flow for this scenario.

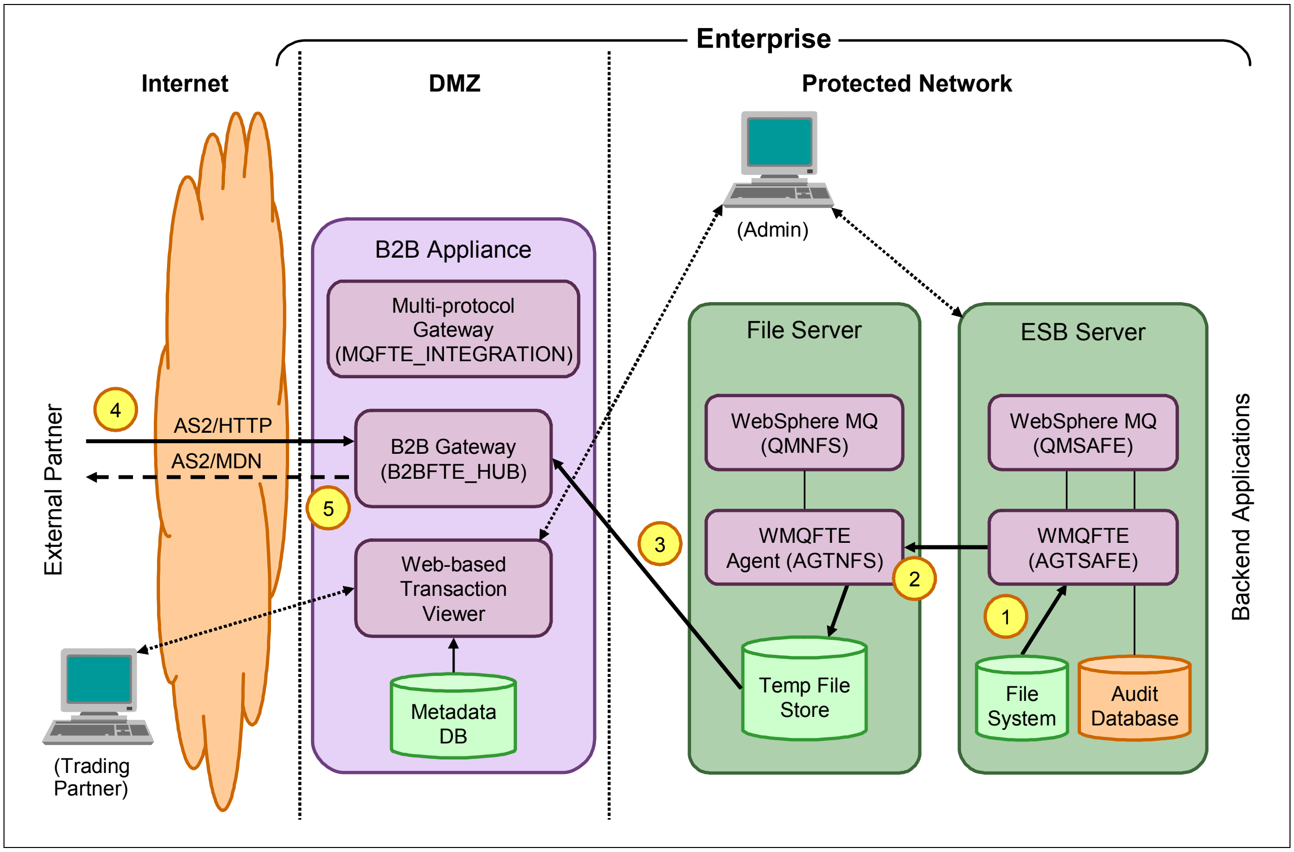


Figure 7-2 Outbound data flow

Figure 7-2 depicts the following process:

1. AGTSAFE reads an EDI X12 file from the file system inside the enterprise and sends the file to AGTNFS.
2. AGTNFS writes the EDI X12 file to the temporary storage location being polled by an NFS poller front-side handler in the B2B Gateway service.
3. The B2B Gateway service, B2BFTE_HUB, retrieves the EDI X12 file from the temporary file storage location. The B2B Gateway service uses profile management to verify and validate the partners, determines the receiving partner's destination, and wraps the EDI X12 file in an AS2 messaging envelope based on attributes (such as MDN request, encryption, signature, and compression) configured for the AS2 destination.
4. The B2BFTE_HUB gateway sends the AS2 message to the receiving partner over HTTP.
5. The partner sends an AS2 MDN to the B2BFTE_HUB gateway, where it is logged, correlated, and persisted by the B2B Gateway service.

7.2.4 Protocols

In this scenario the protocols in use between the external partner and the internal partner on the XB60 are governed by the AS2 standard, which supports HTTP or HTTPS for an application layer protocol. The protocols used for integration with MQ File Transfer Edition are

NFS for file integration and MQ for message level integration. WebSphere MQ File Transfer Edition utilizes both MQ and file system protocols to move files from one location to the next.

7.2.5 Security

The security and partner management of the DataPower XB60 is not a substitute for WebSphere MQ File Transfer Edition security. Use the security of both offerings together to best mitigate risk.

WebSphere MQ File Transfer Edition security

After installation, with no modification, WebSphere MQ File Transfer Edition is unsecured. This might be acceptable for test of evaluation purposes in a protected environment. However, this is not acceptable for production environments.

WebSphere MQ File Transfer Edition offers security features that allow organizations to control who can invoke file transfer operations, who can read and write files being transferred, and how to protect the integrity of files. This can be done through:

- Managing authorities for resources specific to WebSphere MQ File Transfer Edition

  Managing authorities for WebSphere MQ File Transfer Edition resources is done through authorization queues associated with each agent. For any file transfer request, the agent processes require a level of access to their local file systems. Additionally, the user initiating the transfer and user ID running the agent process must have authority to use certain WebSphere MQ objects.

- Managing authorities to access file systems

  The user ID running the agent process must have access to the local file system to read or write files during file transfer. This is controlled through the local operating system.

- Working with sandboxes

  Sandboxing is a term used to define the action of restricting an agent to a certain area of the file system. The area that the agent is allowed to use is its sandbox. For more information about sandboxing, see:

  http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/topic/com.ibm.wmqfte.admin.doc/sandboxes.htm

- Using the agent's commandPath property

  The commandPath property restricts the locations from which an agent can run commands. By default, the commandPath is empty, so an agent cannot call any commands. Take extreme care when this property is set because any command in one of the specified commandPath properties can be called from a remote client system that is able to send commands to the agent. For more information about this property, see:

  http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/topic/com.ibm.wmqfte.admin.doc/command_path.htm

- Configuring SSL encryption for WebSphere MQ File Transfer Edition

  Utilize MQ security to establish an SSL-enabled queue manager.

- Authority to publish log and status messages

  Agents issue various log, progress, and status messages to the coordination queue manager for publication. The publication of these messages can be secured using MQ Security.

For more information about WebSphere MQ File Transfer Edition Security, see the WebSphere MQ File Transfer Edition Information Center:

http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/topic/com.ibm.wmqfte.admin.doc/security_main.htm

WebSphere DataPower B2B Appliance XB60 security

DataPower Appliances meet the toughest security requirements of your networks and have unmatched security assurances with hardware and firmware. The security features of the XB60 are:

- Purpose-built hardware provides physical security
  - Sealed, tamper-evident case
  - No USB ports, external drives, and so on
  - Default *locked-down* configuration
  - Verbose audit log
- Utilization of a hardened firmware image
  - Signed and encrypted by IBM
  - Upgrades in minutes
  - Optimized, embedded DPOS operating system
  - Runs no arbitrary software or Java
- Support for a wide range of security capabilities
  - Authentication, authorization, and auditing (AAA)
  - XML security and threat protection
  - EDIINT AS1, AS2, and AS3 Security (S/MIME)
  - Transport Layer Security (TLS)
  - X.509 certificate management
  - Access control and role-based management
  - Native integration to popular identity management solutions

In this scenario Transport Layer Security such as TLS or SSL is not used because we are using data security provided by the Applicability Statement 2 (AS2) protocol. However, it is considered a best practice to use SSL for external connections whenever the data cannot be encrypted using a standard such as AS2. We use AS signatures and AS encryption, and we request a signed, synchronous message disposition notification (MDN). This requires the use of X.509 certificates.

Additional material: The certificates can be generated by you or you can use the certificates provided in the `B2BScenario_Files\certs` directory in Appendix C, “Additional material” on page 317.

Firewall security

Firewall configuration plays an important role in securing your connections to and from external partners and in protecting the internal network. The external firewall must allow incoming requests from all of the trading partner’s source IP addresses or a range of IP addresses. This can be configured in the outer firewall rules. The method used to configure firewall rules depends on the model and type of firewall being used.

The demilitarized zone (DMZ) is a termination point at the edge of the protected network and is typically used to house internet-facing systems. Setting up tight rules on the inner firewall is important for protecting your internal systems. Typically, inner firewall rules are configured to allow only traffic from a mediation server in the DMZ that terminates the connection from the internet and re-establishes the connection through the inner firewall to a system that the files are destined for or to a system that moves the files to a back-end application.

For this scenario we need to open a Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) ports for 2049 and 111 to support our use of NFS.

7.3 Configuring WebSphere MQ File Transfer Edition

The configuration for WebSphere MQ File Transfer Edition consists of starting the AGTNFS and AGTSAFE agents and setting up a monitor to trigger file transfers from the `C:\B2BFTE\Out` directory to the `/XB60/b2bfte/Out` directory. Files are picked up from the `/XB60/b2bfte/Out` directory by the XB60 and routed to the external partner over AS2.

7.3.1 Security considerations

When trading files with external partners through a B2B gateway product such as the XB60, the security and governance information for the transfers between the external partner and the B2B hub is contained in the configuration and the protocols used to exchange files. The partners do not know anything about or have any access to the WebSphere MQ File Transfer Edition queue managers and agents. To WebSphere MQ File Transfer Edition, the XB60 is the user. A single agent can be used and partners can be identified in the metadata that the XB60 sends to the agent. For securing internal file transfers, WebSphere MQ File Transfer Edition utilizes MQ Security and has added security to control WebSphere MQ File Transfer Edition functionality. However, for this scenario we do not implement WebSphere MQ File Transfer Edition security.

7.3.2 Prerequisites

WebSphere MQ File Transfer Edition has the following prerequisites:

- Installation of WebSphere MQ V7.0 or later
- Installation WebSphere MQ File Transfer Edition V7.0.2 or later

7.3.3 Configuration prerequisites

We assume that the following configuration is in place:

- The following queue managers have been created with the following listener ports:
  - QMNFS (1415) on SYSE
  - QMSAFE (14014) on SYSD

  See “Creating the queue managers” on page 271.
- The WebSphere MQ File Transfer Edition agents, AGTNFS and AGTSAFE. See “Creating the WebSphere MQ File Transfer Edition agents” on page 281.

- The XB60 NFS mount point has been exported on the a file server system. Access to the mount point has been granted for the IP address of the machine running WebSphere MQ File Transfer Edition. The file and directory permissions are set to allow read, write, and execute access.
- Two directories called `/b2bfte/In` and `/b2bfte/out` have been created in the `/XB60` directory on the system running the NFS share. Be sure that directory permissions allow read, write, and execute for everyone.
- Two directories called `C:/B2BFTE/In` and `C:/B2BFTE/Out` have been created on the target system and have been configured to allow read, write, and execute permissions from everyone.

7.3.4 Starting AGTNFS

AGTNFS is used to integrate to the XB60. For this scenario, AGTNFS and the NFS services are running on a Linux system.

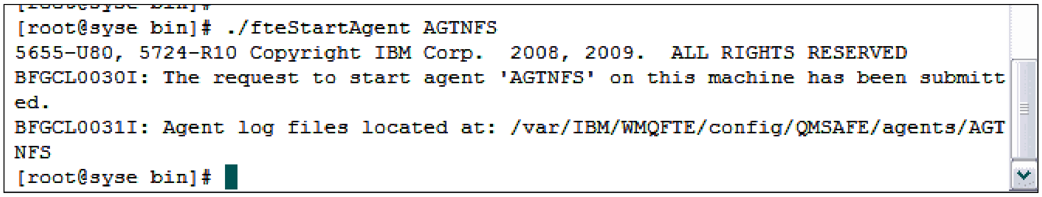
The agent's purpose for the inbound flow is to pick files up from the shared NFS mount point (`/XB60/b2bfte/In`) and transfer them to any other agent and directory that is specified in the XML command file used by the XB60 to trigger the file transfer.

The agent's purpose for the outbound flow is to receive a file from any other agent and to write that file to the shared NFS mount point (`/XB60/b2bfte/Out`) that is being polled by the XB60 for EDI files destined for external partners.

1. To start AGTNFS, log onto the computer running the agent. Navigate to the WebSphere MQ File Transfer Edition bin directory and enter the following command:

   ```
   ./fteStartAgent AGTNFS
   ```

   Figure 7-3 illustrates what the response looks like upon a successful start of the agent.



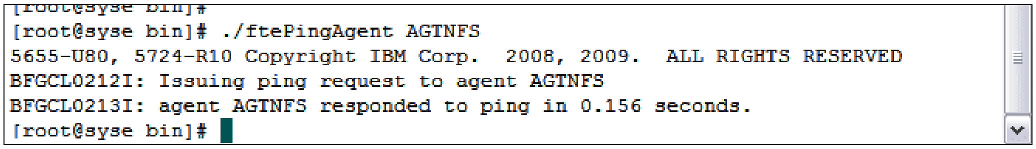
```
[root@syse bin]#
[root@syse bin]# ./fteStartAgent AGTNFS
5655-U80, 5724-R10 Copyright IBM Corp.  2008, 2009.  ALL RIGHTS RESERVED
BFGCL0030I: The request to start agent 'AGTNFS' on this machine has been submitt
ed.
BFGCL0031I: Agent log files located at: /var/IBM/WMQFTE/config/QMSAFE/agents/AGT
NFS
[root@syse bin]# 
```

Figure 7-3   Start AGTNFS

2. Ping AGTNFS to be sure that it is running with the following command:

   ```
   ./ftePingAgent AGTNFS
   ```

   Figure 7-4 illustrates what the response looks like upon a successful ping.



```
[root@syse bin]#
[root@syse bin]# ./ftePingAgent AGTNFS
5655-U80, 5724-R10 Copyright IBM Corp.  2008, 2009.  ALL RIGHTS RESERVED
BFGCL0212I: Issuing ping request to agent AGTNFS
BFGCL0213I: agent AGTNFS responded to ping in 0.156 seconds.
[root@syse bin]# 
```

Figure 7-4   Ping AGTNFS

7.3.5 Starting AGTSAFE

AGTSAFE is the target agent for the inbound flow. It writes the inbound EDI file to the `C:\B2BFTE\In` directory when received from AGTNFS.

For the outbound flow, AGTSAFE is the source agent. It monitors the `C:\B2BFTE\Out` directory for EDI files to send to AGTNFS destined for external partners.

For this scenario, AGTSAFE is running on a Windows system.

1. To start AGTSAFE, log onto the computer running the agent.
2. Enter a command prompt, navigate to the WebSphere MQ File Transfer Edition bin directory, and enter the following command:

   ```
   fteStartAgent AGTSAFE
   ```

   The response is similar to when we started AGTNFS (Figure 7-3).
3. Ping AGTSAFE to be sure that it is running:

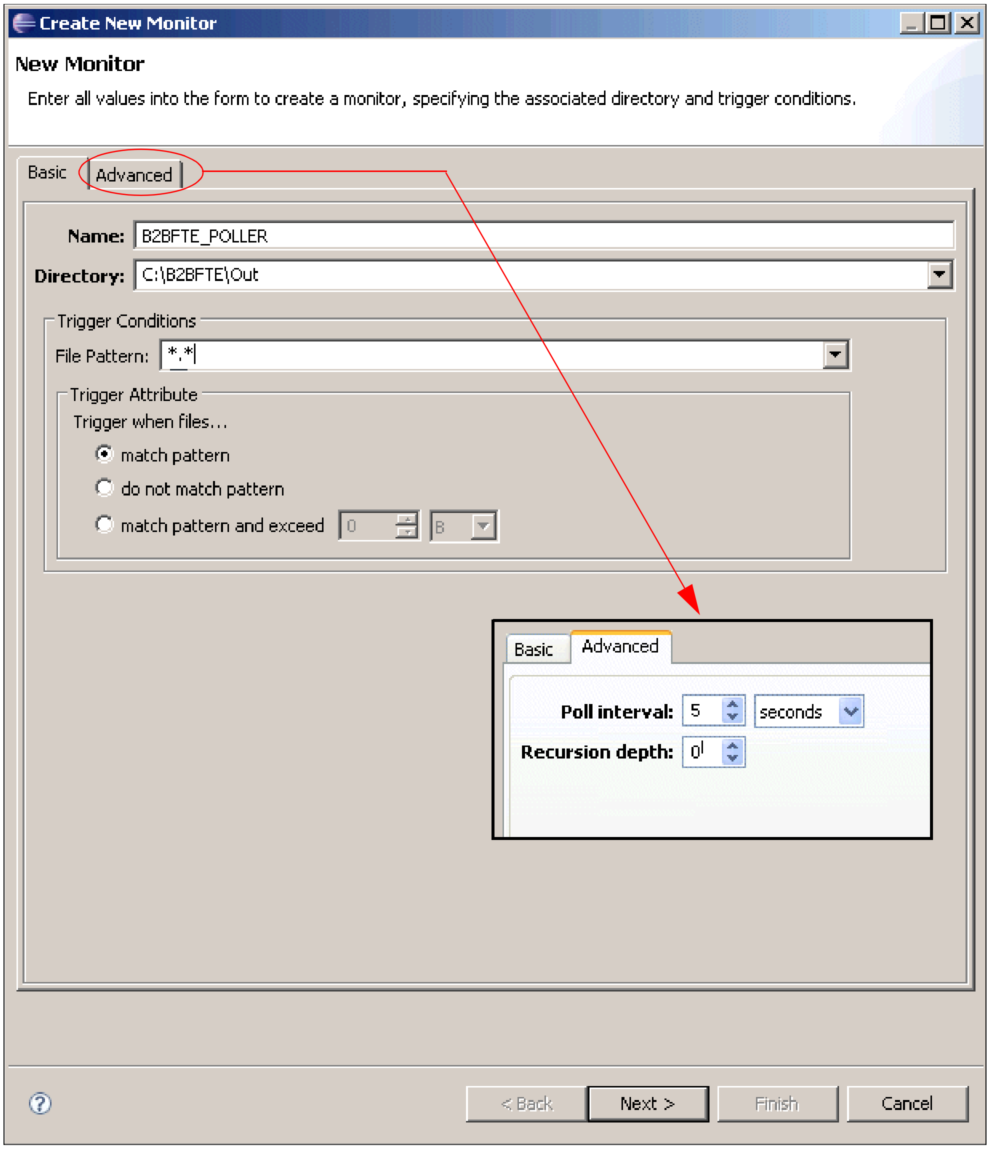
   ```
   ftePingAgent AGTSAFE
   ```

   The response is similar to when we pinged AGTNFS (Figure 7-4 on page 119).

7.3.6 Creating a monitor to poll the outbound directory

For this scenario we want WebSphere MQ File Transfer Edition to monitor the outbound directory that is being used for sending files outbound through AGTNFS to external partners. We use WebSphere MQ Explorer to configure the monitor.

1. Open WebSphere MQ Explorer from the windows Start menu.
2. Navigate to **IBM WebSphere MQ** → **Managed File Transfer** → **QMSAFE** → **Monitors**. Right-click **Monitors** and select **New Monitor**. This launches a new view for configuring the monitor (Figure 7-5).



Create New Monitor

New Monitor

Enter all values into the form to create a monitor, specifying the associated directory and trigger conditions.

Basic | Advanced

Name: B2BFTE_POLLER

Directory: C:\B2BFTE\Out

Trigger Conditions

File Pattern: *.*

Trigger Attribute

Trigger when files...

● match pattern

○ do not match pattern

○ match pattern and exceed 0 B

Basic | Advanced

Poll interval: 5 seconds

Recursion depth: 0

Figure 7-5 Create new file transfer monitor - Trigger

Complete the fields as follows:

a. In the New Monitor - trigger view type `B2BFTE_POLLER` in the Name field.

b. In the Directory field type `C:\B2BFTE\Out`.

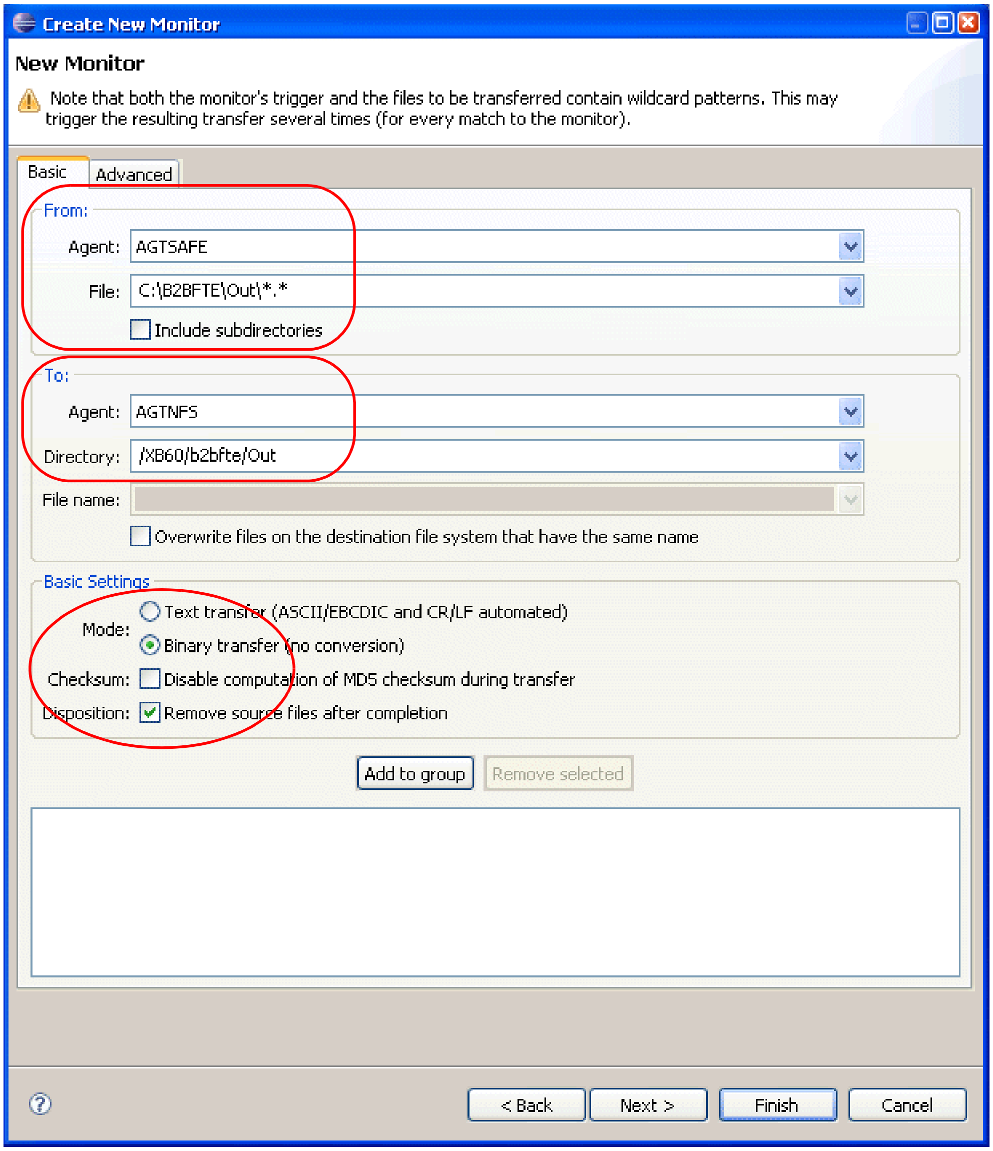
c. In the File Pattern field type `*.*` to pick up any file that gets written to the selected directory.

d. Be sure that the match pattern radio button is selected.

e. Click the **Advanced** tab and configure the value for the desired poll interval. For this scenario we use **5 seconds**.

f. Click **Next** to proceed to the next step.

3. In this step we configure the transfer characteristics as seen in Figure 7-6.



New Monitor

Note that both the monitor's trigger and the files to be transferred contain wildcard patterns. This may trigger the resulting transfer several times (for every match to the monitor).

From:

Agent: AGTSAFE

File: C:\B2BFTE\Out\*.*

□ Include subdirectories

To:

Agent: AGTNFS

Directory: /XB60/b2bfte/Out

File name:

□ Overwrite files on the destination file system that have the same name

Basic Settings

Mode: ○ Text transfer (ASCII/EBCDIC and CR/LF automated)

● Binary transfer (no conversion)

Checksum: □ Disable computation of MD5 checksum during transfer

Disposition: ☑ Remove source files after completion

Figure 7-6 Create new file transfer monitor - To and from agents

Complete the fields as follows:

a. In the From section use the drop-down menu to select **AGTSAFE** in the Agent field.

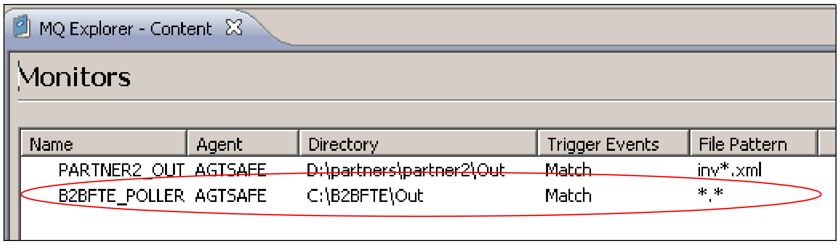
b. In the From section, in the File field enter `C:\B2BFTE\Out\`.

c. In the To section use the drop-down menu to select **AGTNFS** in the Agent field.

d. In the To File field enter `/XB60/b2bfte/Out`.

   e. Leave Binary transfer selected in the Basic Settings section and select **Remove source files after completion** in the Disposition field. Click **Next** to continue to the next step.

4. We do not send any metadata to the XB60 in this scenario, so we can skip this step. Click **Next** to continue.
5. In the Transfer Summary view click **Finish** to complete the monitor configuration.
6. Refresh WebSphere MQ Explorer and you will see a new monitor called B2BFTE_POLLER (Figure 7-7).



Monitors

Name	Agent	Directory	Trigger Events	File Pattern
PARTNER2_OUT	AGTSAFE	D:\partners\partner2\Out	Match	inv*.xml
B2BFTE_POLLER	AGTSAFE	C:\B2BFTE\Out	Match	*.*

Figure 7-7 Configured Monitors view

7.4 Configuring WebSphere DataPower B2B Appliance XB60

This section describes the configuration steps required in the WebSphere DataPower B2B Appliance to implement the scenario flows as depicted in Figure 7-1 on page 114 and Figure 7-2 on page 115:

- Configuration of an NFS static mount
- Configuration of an MQ queue manager object
- Configuration of a multi-protocol gateway service for integrating to WebSphere MQ File Transfer Edition
- Configuration of Trading Partner Profiles
- Configuration of a B2B Gateway service for trading AS2 messages

The installation of the XB60 is not covered in this book. For information about how to install the appliance, see the installation documentation. For the purposes of this scenario, we configure a B2B gateway in a single domain that acts as your B2B hub. The partner's B2B hub is preconfigured in its own domain to allow us to simulate an external partner without the need for another machine. We assume that you have the proper credentials to log on to the XB60 and perform the configuration.

7.4.1 Software and hardware prerequisites

The prerequisite for XB60 is WebSphere DataPower B2B Appliance Model 9235-62X with firmware Version 3.8.1.2 or later.

7.4.2 Configuration prerequisites

We assume that the following configuration is in place:

- The WebSphere MQ File Transfer Edition agents, AGTNFS and AGTSAFE, are running.
- You have valid user credentials on the XB60.
- On the XB60, an application domain other then default has been created for this scenario.

7.4.3 Configuring a static NFS mount object

The static NFS mount is needed to support the connection to the NFS directory that is shared between AGTNFS and the XB60. This mount point is used by the multi-protocol gateway policy being used for MQ File Transfer Edition integration.

To configure the Static NFS Mount object use the following the steps:

1. From the left navigation menu in the XB60, select **Objects** → **Network Settings** → **NFS Static Mounts** (Figure 7-8).

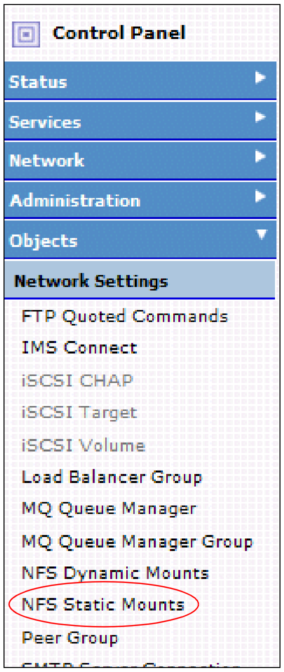
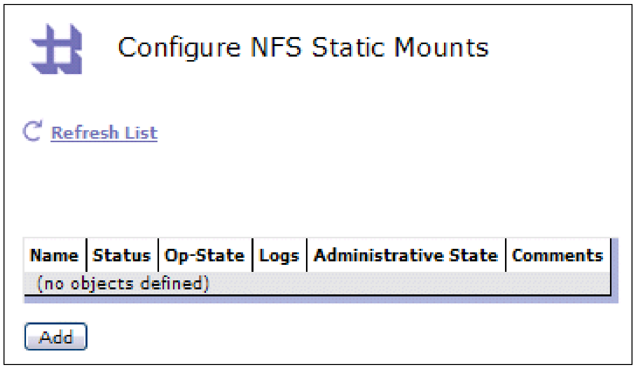


Figure 7-8 Left navigation menu - NFS

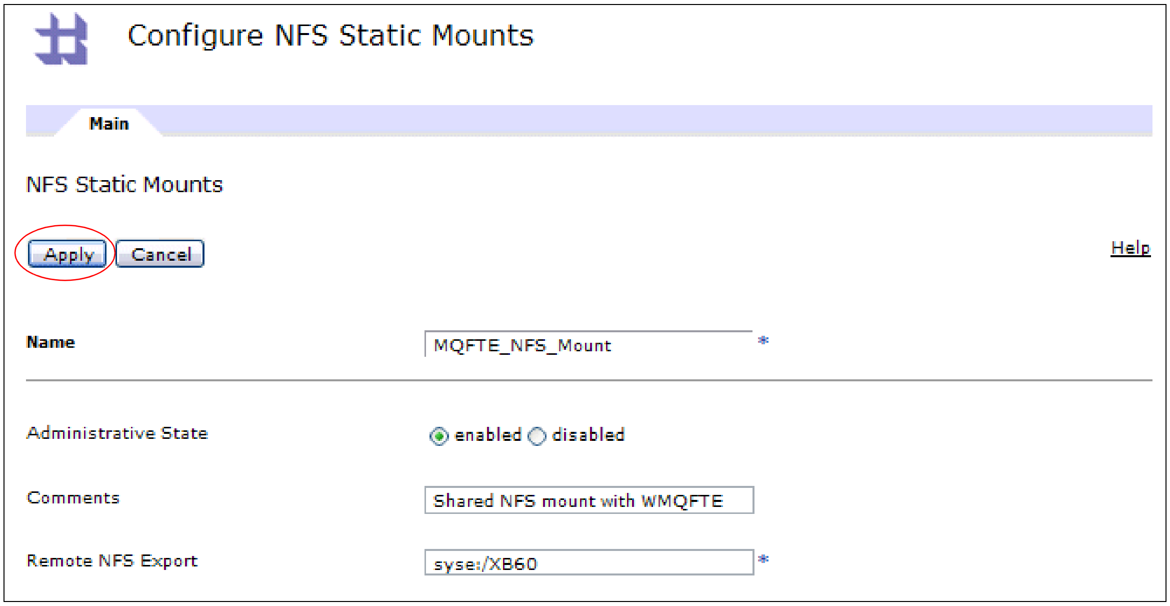
2. In the Configure NFS Static Mounts list view, click **Add** (Figure 7-9 on page 124).



Name	Status	Op-State	Logs	Administrative State	Comments
(no objects defined)					

Figure 7-9 Configure NFS Static Mounts list view

3. Figure 7-10 illustrates how the configured Static NFS Mount object should look. Fields that are populated with default values might not be displayed.



Name	MQFTE_NFS_Mount *
Administrative State	● enabled ○ disabled
Comments	Shared NFS mount with WMQFTE
Remote NFS Export	syse:/XB60 *

Figure 7-10 NFS Static Mount object

Complete the fields as follows:

a. On the Main tab enter `MQFTE_NFS_Mount` in the Name field and make sure that the administrative state is **enabled**.

b. In the Remote NFS Export field enter the IP address or host name of the file server followed by the directory path. For the purpose of this scenario use `syse:/XB60`.

c. Leave all remaining fields with the default values, and then click **Apply**.

4. Save the configuration by clicking **Save Config** in the upper right corner of the page.

> **Troubleshooting tip:** You might see that the object is down because the process is carried out asynchronously. Click **Configure NFS Static Mounts** at the very top of this view and you will be taken back to the list view, where you should see that the mount is now up. If the object still does not come up, you might have the wrong values in the Remote NFS Export field, or the host, NFS, or both, are not accessible. Check to be sure that the NFS mount is up, the directory rights are set correctly, and the host is accessible from the XB60. You can ping the host on the DataPower Troubleshooting page, which is accessible from the Control Panel.

7.4.4 Configuring an MQ Queue Manager object

The MQ Queue Manager object is needed to support our connection to the WebSphere MQ File Transfer Edition command queue, which is used to consume the XML command message that triggers the file transfer between AGTNFS and AGTSAFE. This queue manager information is used by the multi-protocol gateway processing policy.

To configure the MQ Queue Manager object use the following the steps:

1. From the left navigation menu select **Network** → **Other** → **MQ Queue Manager** (Figure 7-11).

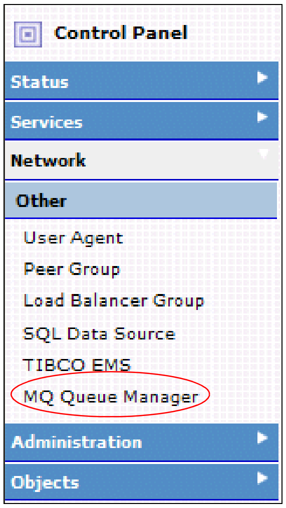
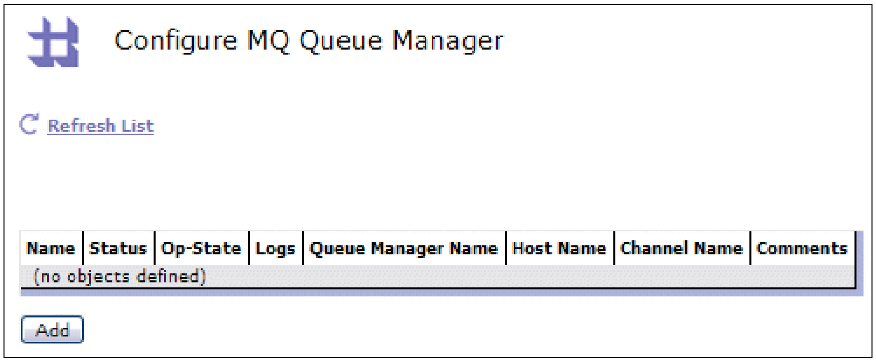


Figure 7-11 Left navigation menu - MQ Queue Manager

2. In the Configure MQ Queue Manager list view, click **Add**.

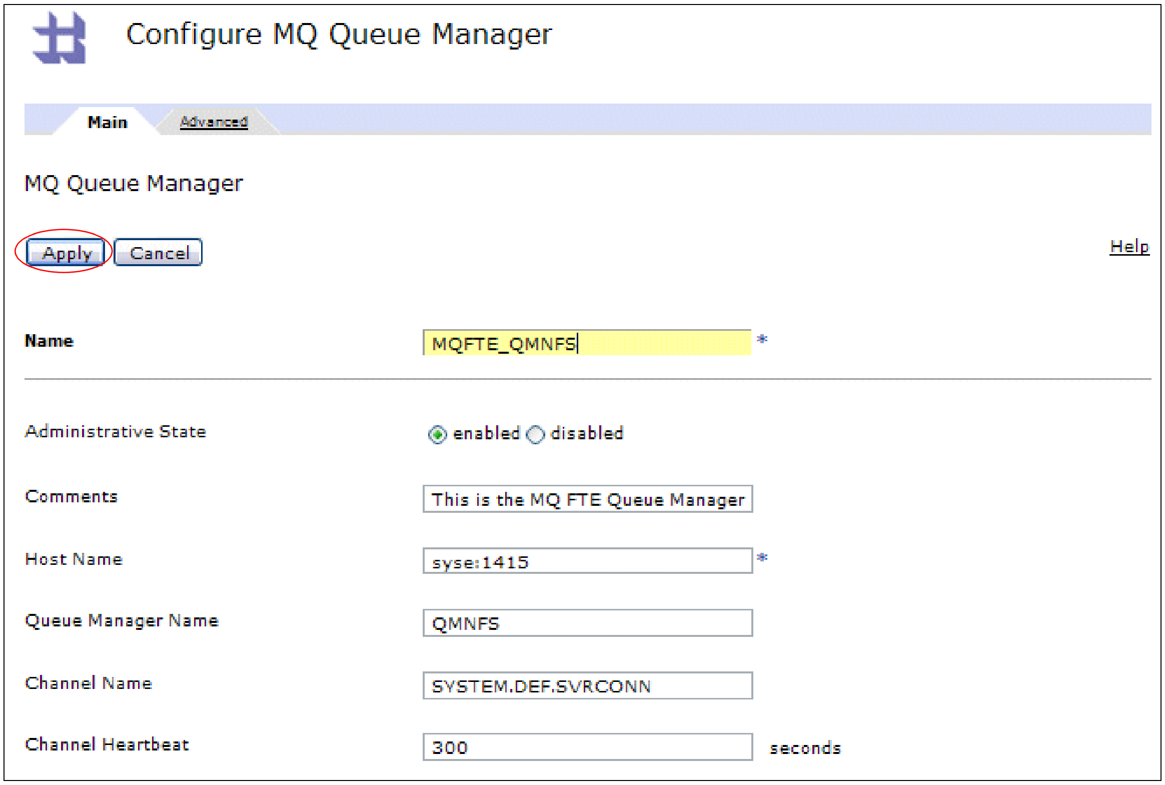

Configure MQ Queue Manager

Refresh List

Name	Status	Op-State	Logs	Queue Manager Name	Host Name	Channel Name	Comments
(no objects defined)							

Figure 7-12 Configure Queue Manager list view

3. Figure 7-13 illustrates how the configured MQ Queue Manager object should look. Fields that are populated with default values might not be displayed. In the Configure MQ Queue Manager view, the first tab is the Main tab, and it is the only tab that we configure. This scenario does not require the use of the other tab.



MQ Queue Manager

Name	MQFTE_QMNFS *
Administrative State	● enabled ○ disabled
Comments	This is the MQ FTE Queue Manager
Host Name	syse:1415 *
Queue Manager Name	QMNFS
Channel Name	SYSTEM.DEF.SVRCONN
Channel Heartbeat	300 seconds

Figure 7-13 MQ Queue Manager object

Complete the fields as follows:

a. Enter `MQFTE_QMNFS` in the Name field and make sure that the administrative state is **enabled**.

b. In the Host Name field enter the IP address or host name of the server running the queue manager followed by the port number. For the purpose of this scenario use `syse:1415`.

c. In the Queue Manager Name field type `QMNFS`.

d. Leave all remaining fields with the default values, and then click **Apply**.

4. Save the configuration by clicking **Save Config** in the upper right corner of the page.

> **Troubleshooting tip:** You might see that the object will be down because the process is carried out asynchronously. Click **Configure MQ Queue Manager** at the very top of this view and you will be taken back to the list view, where you should see that the mount is now up. If the object still does not come up, you might have the wrong values in the Host Name, Port, or Queue manager fields, or the host, MQ, or both, is not accessible. Check to be sure that the queue manager is up and that the host is accessible from the XB60. You can ping the host in the DataPower Troubleshooting page, which is accessible from the Control Panel.

7.4.5 Configuring a multi-protocol gateway for WebSphere MQ File Transfer Edition integration

The multi-protocol gateway is used to integrate to WebSphere MQ File Transfer Edition through a shared NFS mount point. The gateway uses a custom action to write the file to the file system with a unique file name and to generate and send an XML command message that triggers the file transfer.

1. From the left navigation menu of the XB60, select **Services** → **New Multi-Protocol Gateway** (Figure 7-14).

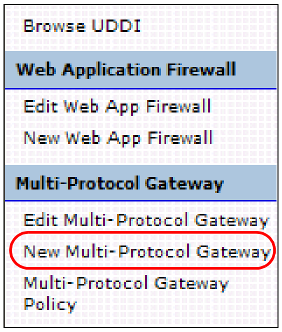
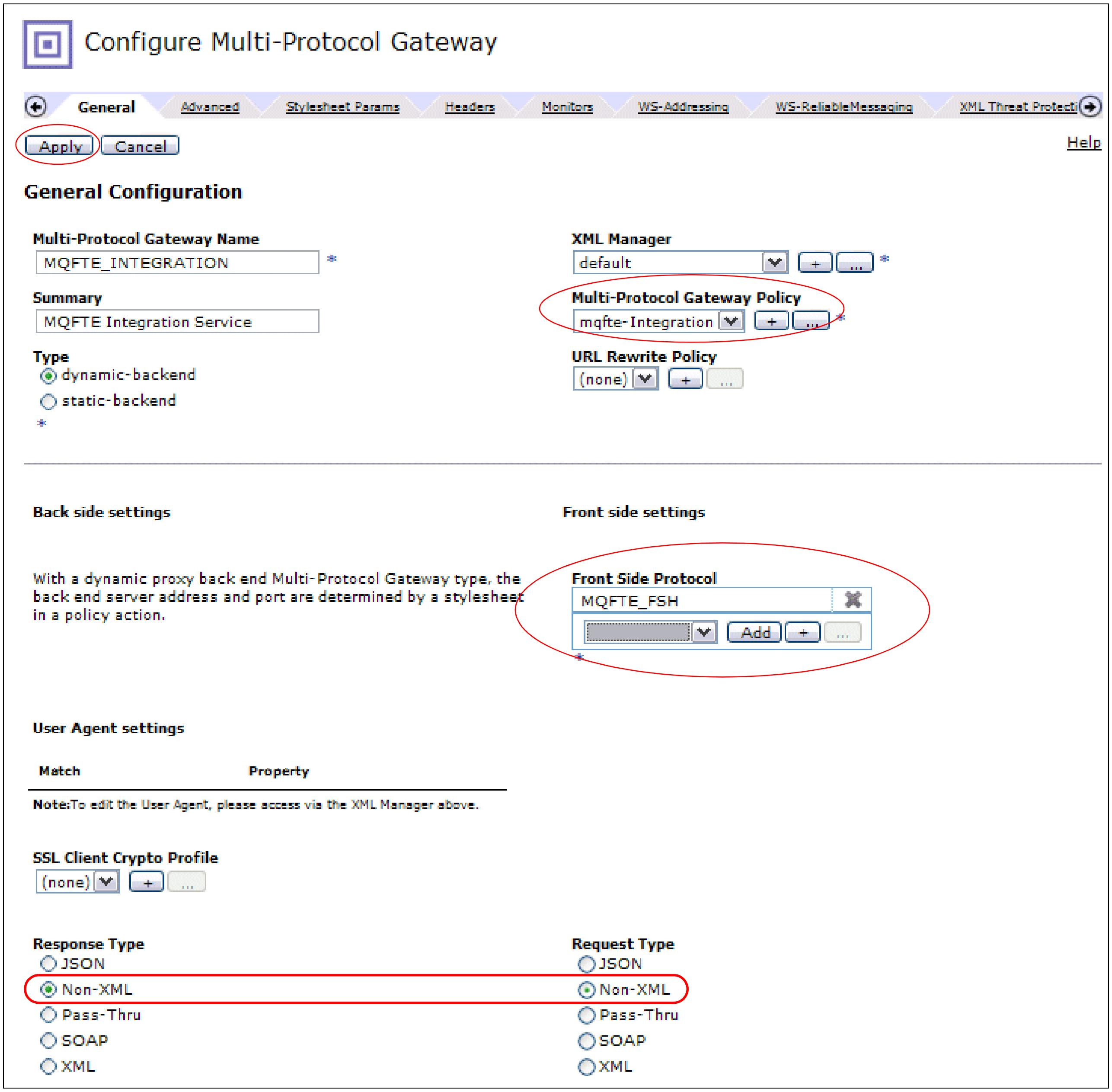


Figure 7-14 Left navigation menu - Multi-Protocol Gateway

2. In the Configure Multi-Protocol Gateway view the first tab is the General tab and it is the only tab that we configure. This scenario does not require the use of the other tabs (Figure 7-15).



Configure Multi-Protocol Gateway

General Configuration

Multi-Protocol Gateway Name: MQFTE_INTEGRATION *

Summary: MQFTE Integration Service

Type
● dynamic-backend
○ static-backend
*

XML Manager: default *

Multi-Protocol Gateway Policy: mqfte-Integration *

URL Rewrite Policy: (none)

Back side settings

With a dynamic proxy back end Multi-Protocol Gateway type, the back end server address and port are determined by a stylesheet in a policy action.

Front side settings

Front Side Protocol: MQFTE_FSH
*

User Agent settings

Match	Property

Note:To edit the User Agent, please access via the XML Manager above.

SSL Client Crypto Profile: (none)

Response Type
○ JSON
● Non-XML
○ Pass-Thru
○ SOAP
○ XML

Request Type
○ JSON
● Non-XML
○ Pass-Thru
○ SOAP
○ XML

Figure 7-15 Configure Multi-Protocol Gateway

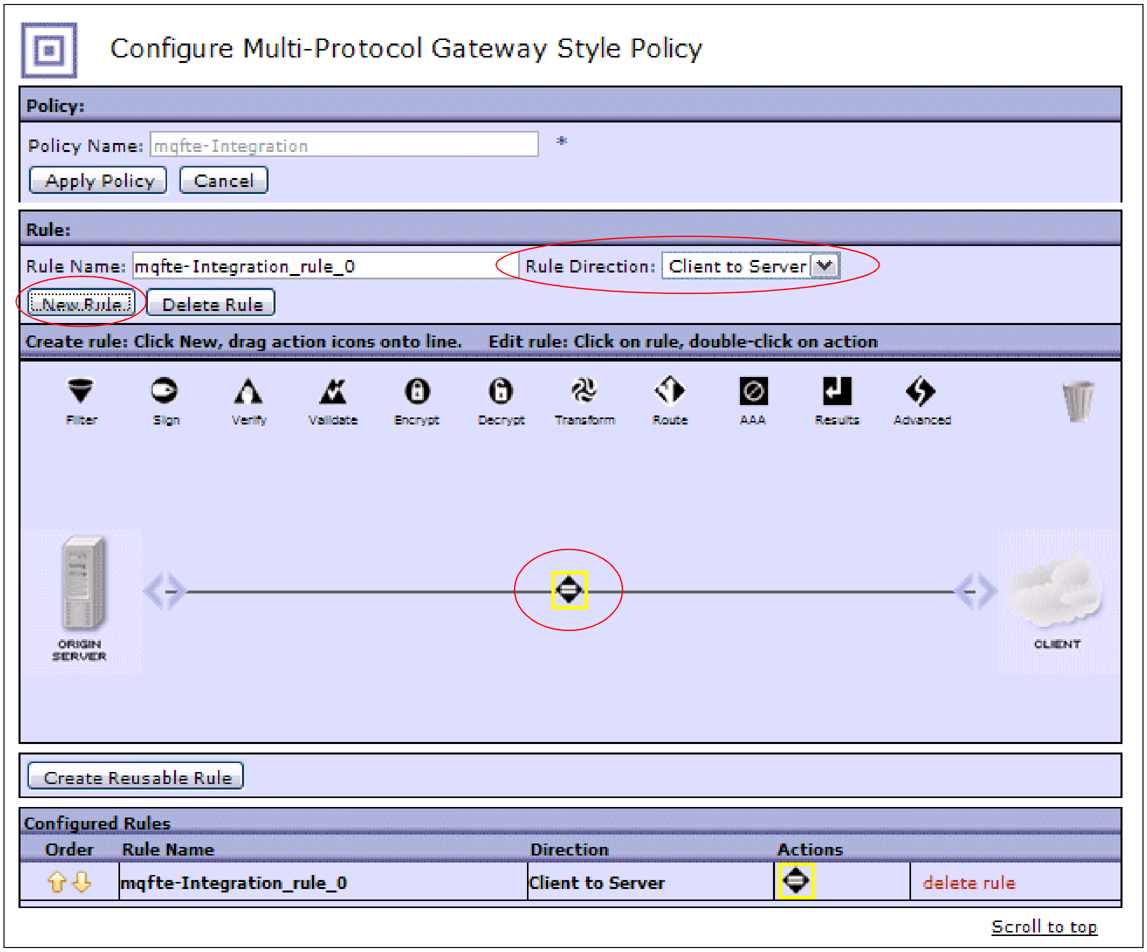
Complete the fields as follows:

a. In the Multi-Protocol Gateway Name field enter `MQFTE_INTEGRATION`.

b. In the Type field select **dynamic-backend**.

c. Set the Request Type and Response Type fields to **Non-XML**.

3. Create a multi-protocol gateway policy by clicking the Plus icon (+) next to the drop-down menu (circled in Figure 7-15 on page 129). This opens the Configure Multi-Protocol Gateway Style Policy window (Figure 7-16).



Policy Name: mqfte-Integration *

Rule Name: mqfte-Integration_rule_0

Rule Direction: Client to Server

Create rule: Click New, drag action icons onto line. Edit rule: Click on rule, double-click on action

Configured Rules

Order	Rule Name	Direction	Actions	
⇧⇩	mqfte-Integration_rule_0	Client to Server	◆	delete rule

Figure 7-16 Configure Multi-Protocol Gateway Policy

Complete the fields as follows:

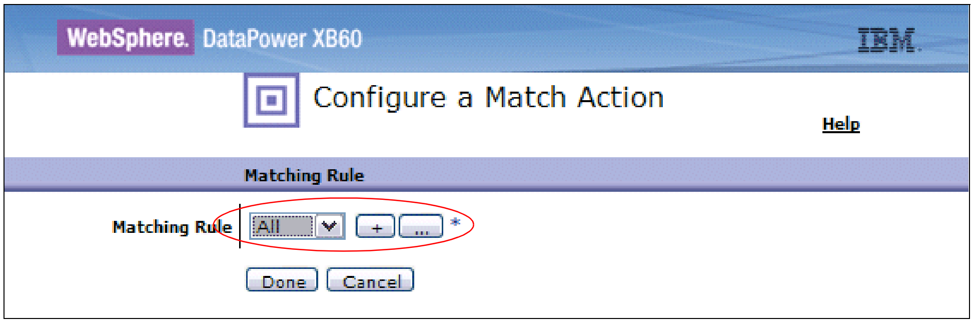
a. In the Configure Multi-Protocol Gateway Policy view, enter `mqfte-integration` in the Policy Name field.

b. Click **New Rule** to create the first rule. The rule name is filled in for you and the Match action is automatically added to the pallet (Figure 7-16).

c. Change the rule direction to **Client to Server**.

d. Double-click the Match Action icon (◆) to configure it.

   i. The Configure a Match Action view is displayed (Figure 7-17). Use the drop-down menu in the Matching Rule field to select **All**.



Matching Rule: All *

Figure 7-17 Configure Match Action view

> **Attention:** If All is not displayed in the drop-down menu it must be created by following steps ii through viii. If All is displayed select it and skip to step ix.

   ii. Click the Plus icon next to the Matching Rule field.

   iii. Click the **Matching Rule** tab and type `All` in the Name field.

   iv. Click **Add** to add a Matching Rule.

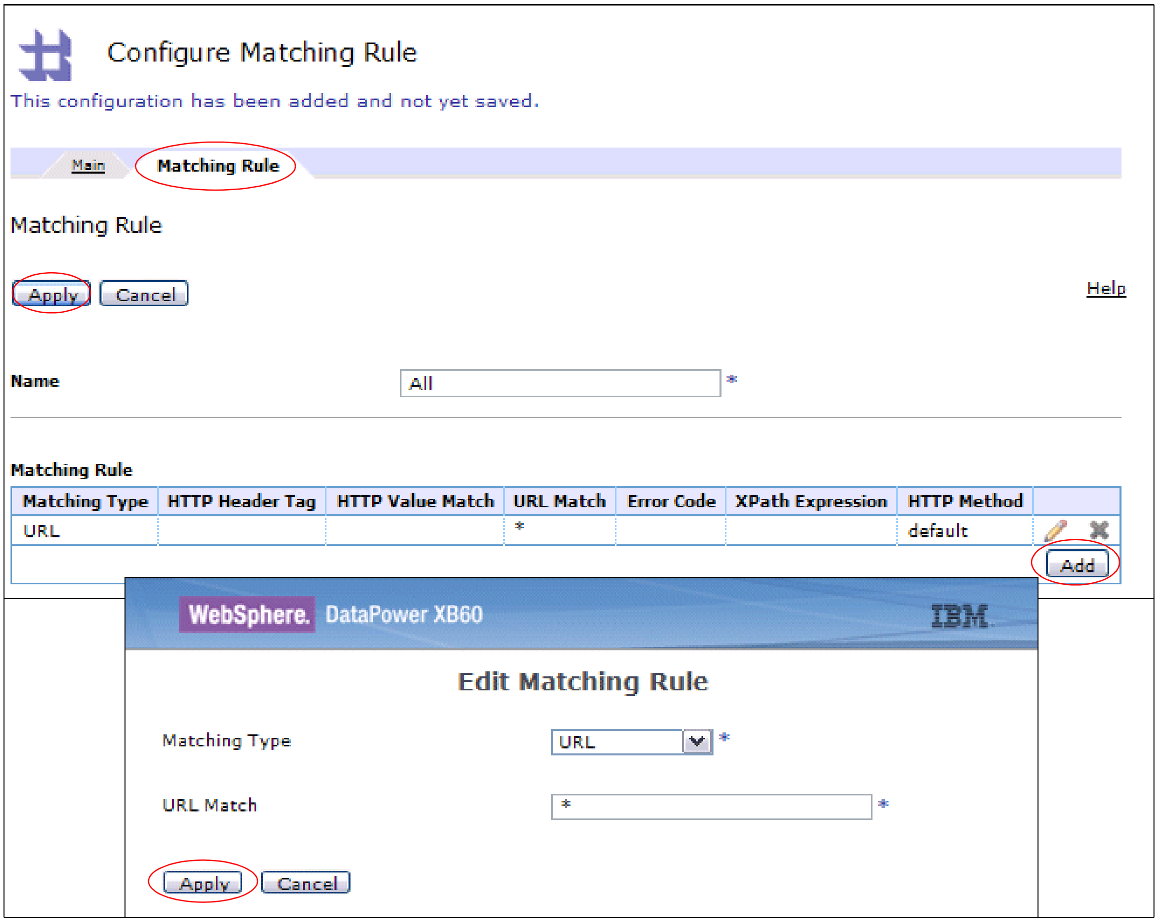
   v. Select **URL** in the Matching Type field.

   vi. Type an asterisk (*) in the URL Match field.

   vii. Click **Apply** in the Edit Matching Rule view.

viii.Click **Apply** in the Configure Matching Rule view.

Figure 7-18 illustrates the steps and how the configured matching rule should look. Fields that are populated with default values might not be displayed.



Configure Matching Rule

This configuration has been added and not yet saved.

Matching Rule

Name: All *

Matching Rule

Matching Type	HTTP Header Tag	HTTP Value Match	URL Match	Error Code	XPath Expression	HTTP Method
URL			*			default

Edit Matching Rule

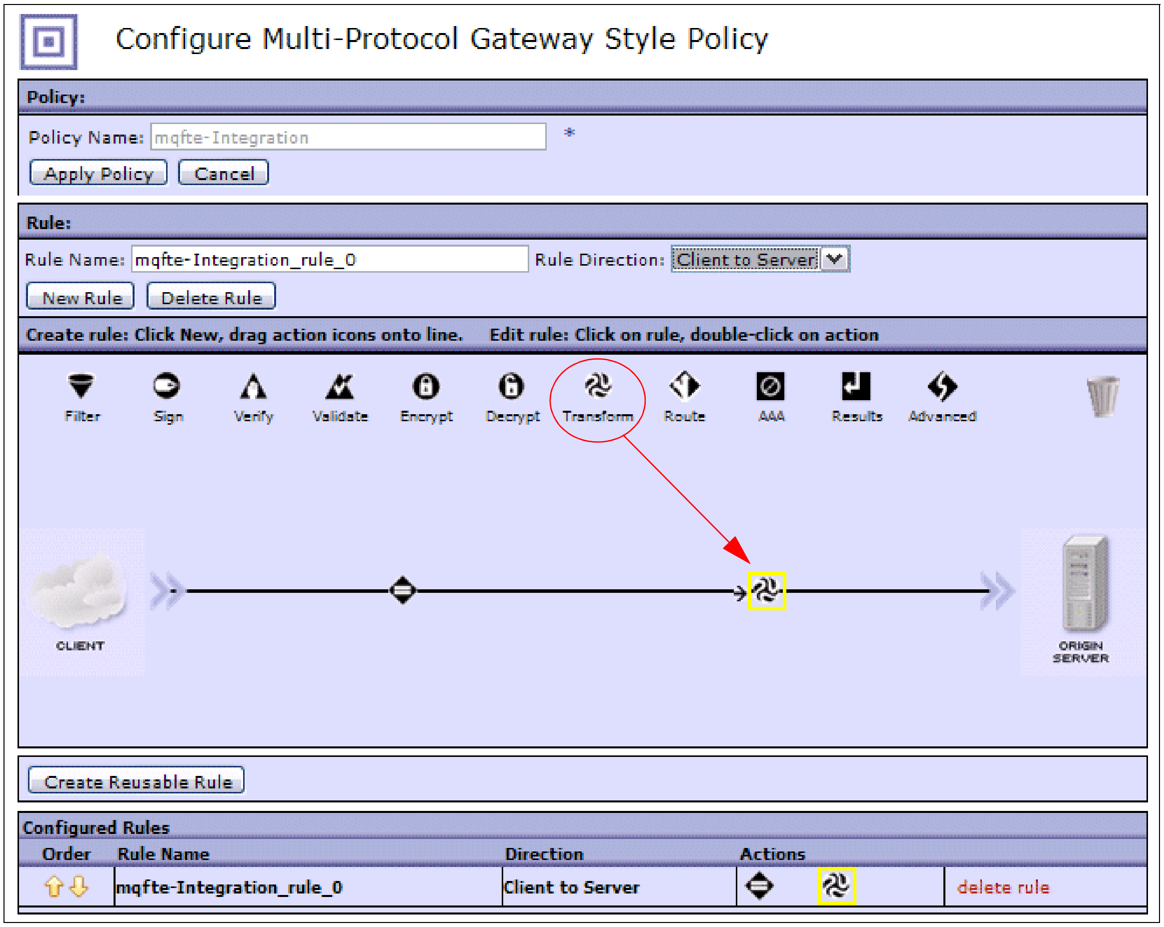
Matching Type: URL *

URL Match: * *

Figure 7-18 Configure matching rule

ix. Click **Done** in the Configure a Match Action view (Figure 7-17 on page 131) and you are returned to the Configure Multi-Protocol Gateway Style Policy view.

e. Drag a **Transform Action** on to the palette to the right of the Match action (Figure 7-19).



Configure Multi-Protocol Gateway Style Policy

Policy:

Policy Name: mqfte-Integration *

Rule:

Rule Name: mqfte-Integration_rule_0 Rule Direction: Client to Server

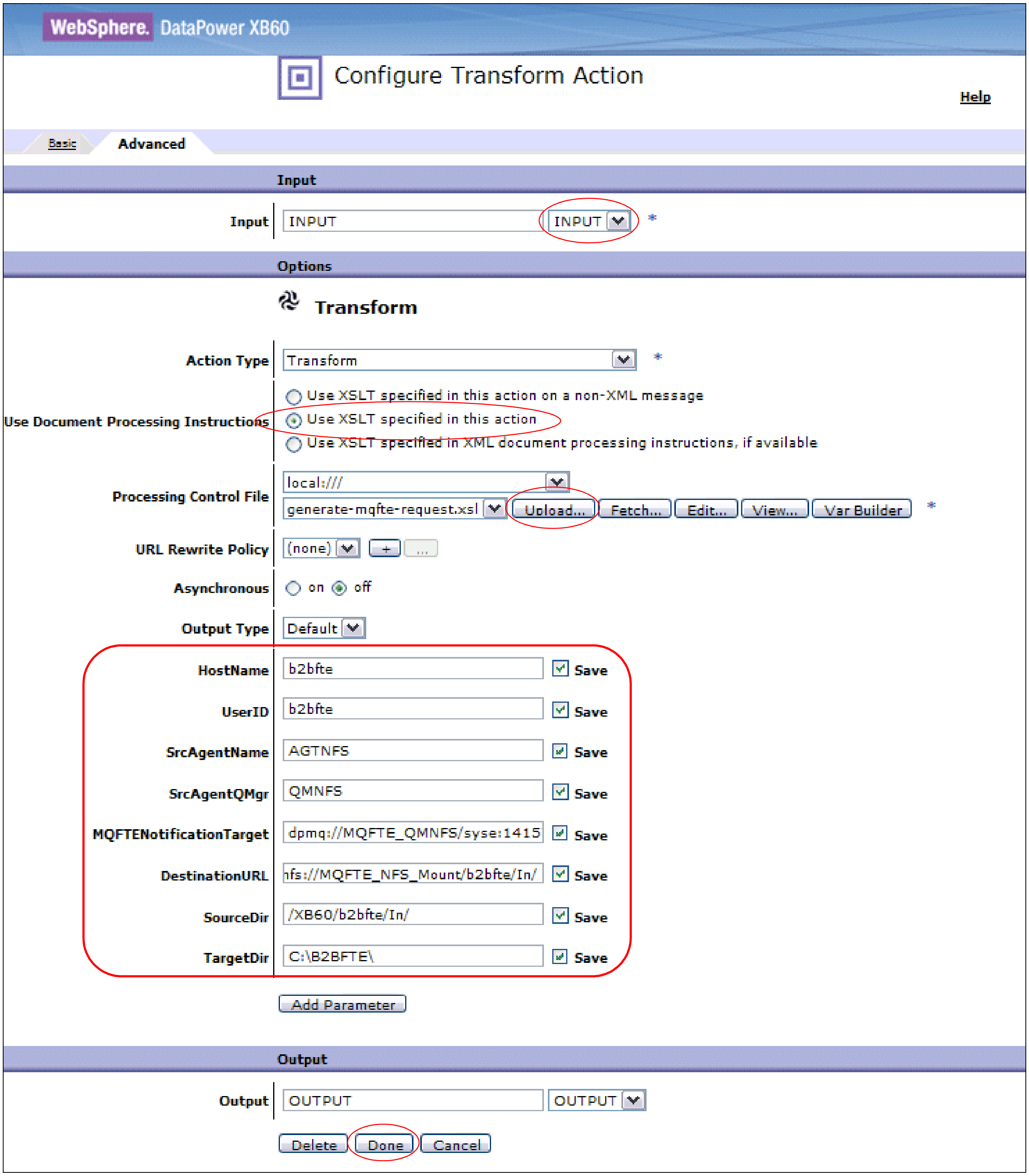
Create rule: Click New, drag action icons onto line. Edit rule: Click on rule, double-click on action

Configured Rules

Order	Rule Name	Direction	Actions	
⇧⇩	mqfte-Integration_rule_0	Client to Server		delete rule

Figure 7-19 Multi-Protocol Gateway Style Policy - Transform action

f. Double-click **Transform Action** to configure it. Figure 7-20 illustrates how the configured Transform action should look. Fields that are populated with default values might not be displayed.



Configure Transform Action

Input

Input: INPUT | INPUT *

Options

Transform

Action Type: Transform *

Use Document Processing Instructions:
○ Use XSLT specified in this action on a non-XML message
● Use XSLT specified in this action
○ Use XSLT specified in XML document processing instructions, if available

Processing Control File: local:/// generate-mqfte-request.xsl *

URL Rewrite Policy: (none)

Asynchronous: ○ on ● off

Output Type: Default

HostName	b2bfte	☑ Save
UserID	b2bfte	☑ Save
SrcAgentName	AGTNFS	☑ Save
SrcAgentQMgr	QMNFS	☑ Save
MQFTENotificationTarget	dpmq://MQFTE_QMNFS/syse:1415	☑ Save
DestinationURL	nfs://MQFTE_NFS_Mount/b2bfte/In/	☑ Save
SourceDir	/XB60/b2bfte/In/	☑ Save
TargetDir	C:\B2BFTE\	☑ Save

Output

Output: OUTPUT | OUTPUT

Figure 7-20 Configure Transform action

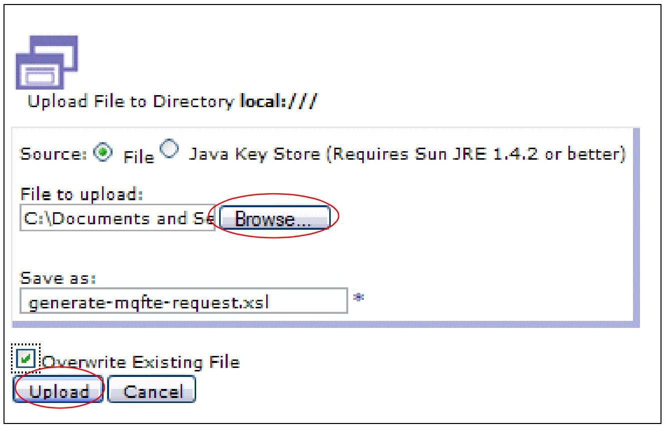
To configure these fields:

i. Click the **Advanced** tab, click the drop-down menu for the Input field, and then select **INPUT**.

ii. Be sure that **Use XSLT specified in this action** is selected in the Use Document Processing Instructions field and click **Upload** to upload the XSL that we are using for this policy.

iii. In the Upload File view click **Browse**, navigate to the `generate-mqfte-request.xsl` file, and then click **Open**.

> **Additional material:** The `generate-mqfte-request.xsl` is located in the `Common_Files` directory in the download materials.



Upload File to Directory **local:///**

Source: ● File ○ Java Key Store (Requires Sun JRE 1.4.2 or better)

File to upload:
C:\Documents and Se

Save as:
generate-mqfte-request.xsl *

☑ Overwrite Existing File

Figure 7-21 Upload XSL File

iv. Be sure that **Overwrite Existing File** is selected, click **Upload**, and then click **Continue** in the confirmation box to return to the Advanced tab.

v. Configure the parameters used in the `generate-mqfte-request.xsl` processing control file using Table 7-1.

Table 7-1 generate-mqfte-request.xsl parameters

Parameter name	**Value**
HostName	`b2bfte`
UserID	`b2bfte`
SrcAgentName	`AGTNFS`
SrcAgentQMgr	`QMNFS`
MQFTENotificationTarget	`dpmq://MQFTE_QMNFS/syse:1415?RequestQueue=SYSTEM.FTE.COMMAND.AGTNFS;SetReplyTo=true`
DestinationURL	`dpnfs://MQFTE_NFS_Mount/b2bfte/In/`
SourceDir	`/XB60/b2bfte/In/`
TargetDir	`C:\B2BFTE\In\`

vi. Click the drop-down menu for the Output field and select **OUTPUT**.

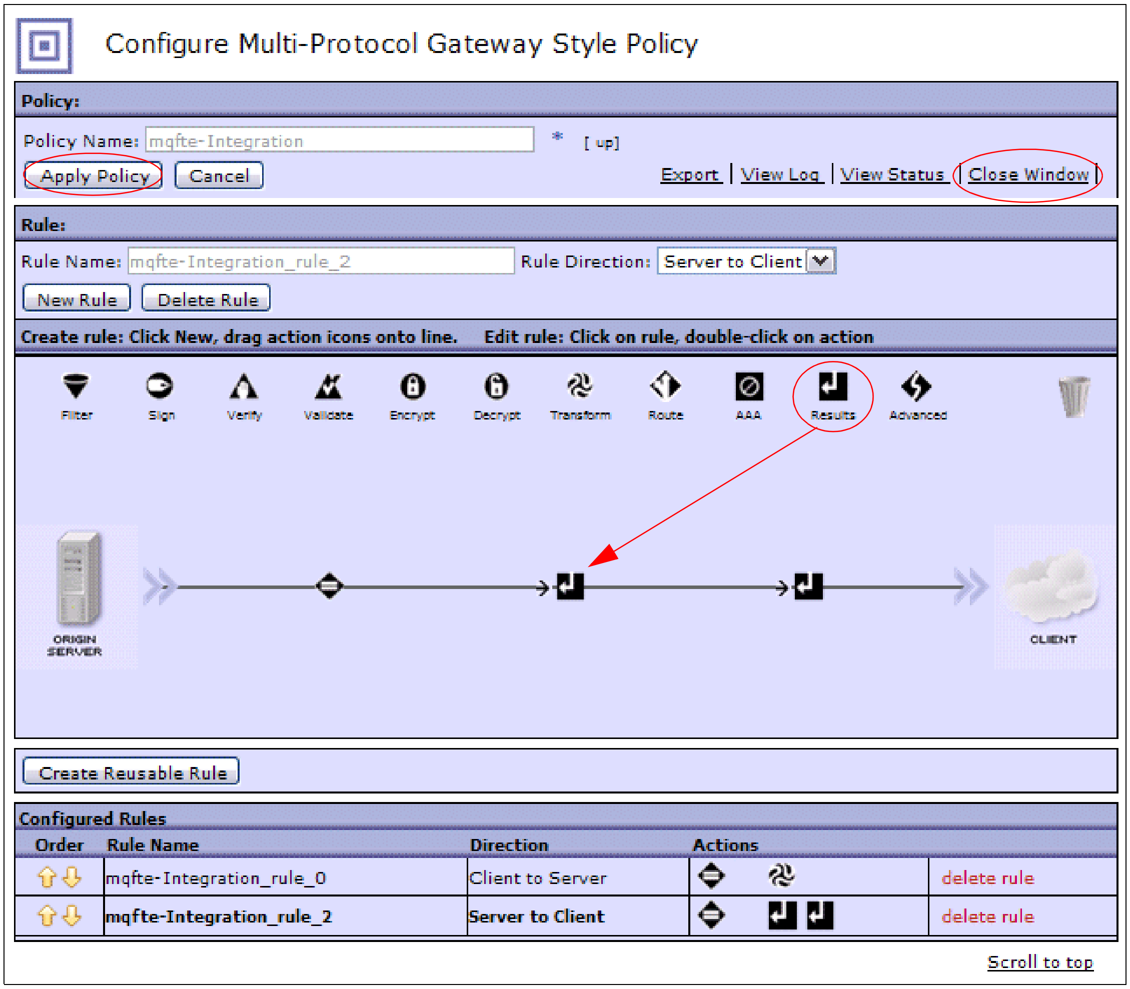
vii. Leave the default values in all other fields and click **Done** to complete the configuration of the Transform action, returning you to the Configure Multi-Protocol Gateway Style Policy view.

g. Click **New Rule** to create the second rule to be used for the response. The rule name is filled in for you and the Match action is automatically added to the pallet.

i. Change the rule direction to **Server to Client**.

ii. Double-click the **Match Action** and use the drop-down menu to set it to **All**, and then click **Done**.

h. Drag a **Results** action onto the palette to the right of the Match action (Figure 7-22).



Configure Multi-Protocol Gateway Style Policy

Policy:

Policy Name: mqfte-Integration * [up]

Apply Policy | Cancel | Export | View Log | View Status | Close Window

Rule:

Rule Name: mqfte-Integration_rule_2 Rule Direction: Server to Client

New Rule | Delete Rule

Create rule: Click New, drag action icons onto line. Edit rule: Click on rule, double-click on action

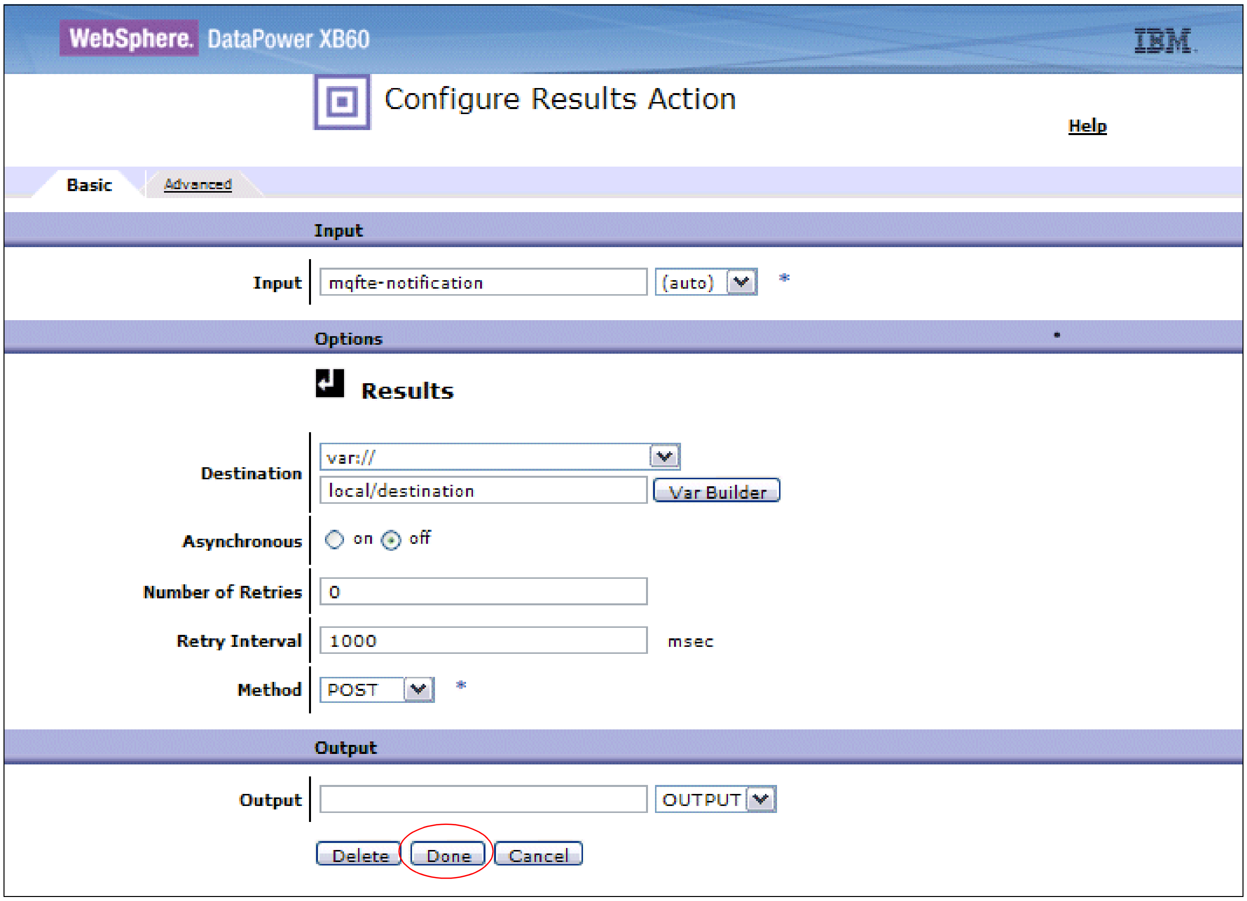
Configured Rules

Order	Rule Name	Direction	Actions	
⇧⇩	mqfte-Integration_rule_0	Client to Server	◆ ✲	delete rule
⇧⇩	**mqfte-Integration_rule_2**	**Server to Client**	◆ ↵ ↵	delete rule

Figure 7-22 Multi-Protocol Gateway Style Policy - Configure Results action

Configure the Results action as follows:

i. Double-click **Results Action** to configure it. Figure 7-23 illustrates how the configured Results action should look. Fields that are populated with default values might not be displayed.



Configure Results Action

Input

Input: mqfte-notification (auto) *

Options

Results

Destination: var://
local/destination

Asynchronous: ○ on ● off

Number of Retries: 0

Retry Interval: 1000 msec

Method: POST *

Output

Output: OUTPUT

Figure 7-23 Configure Results action

iii. In the Configure Results Action view on the Basic tab type `mqfte-notification` in the Input field.

iv. In the Destination field click the drop-down menu, select **var://**, and type `local/destination`.

v. Use the default values in the remaining fields and click **Done**, which returns you to the Configure Multi-Protocol Gateway Style Policy view.

j. Click **Apply Policy** in the Configure Multi-Protocol Gateway Style Policy view.

> **Second Results action:** After clicking **Apply Policy**, a second result action is added to the right of the first Results action. This is normal. No configuration of this action is needed.

k. Close the Configure Multi-Protocol Gateway Style Policy view by clicking **Close Window**, which returns you to the Configure Multi-Protocol view (Figure 7-15 on page 129).

4. Create an HTTP front-side protocol handler for the multi-protocol gateway by clicking the Plus icon in the Front Side Protocol field. Choose **HTTP Front Side Handler** from the list.

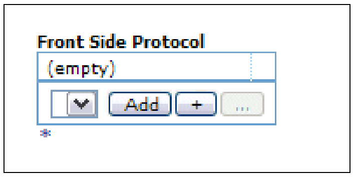
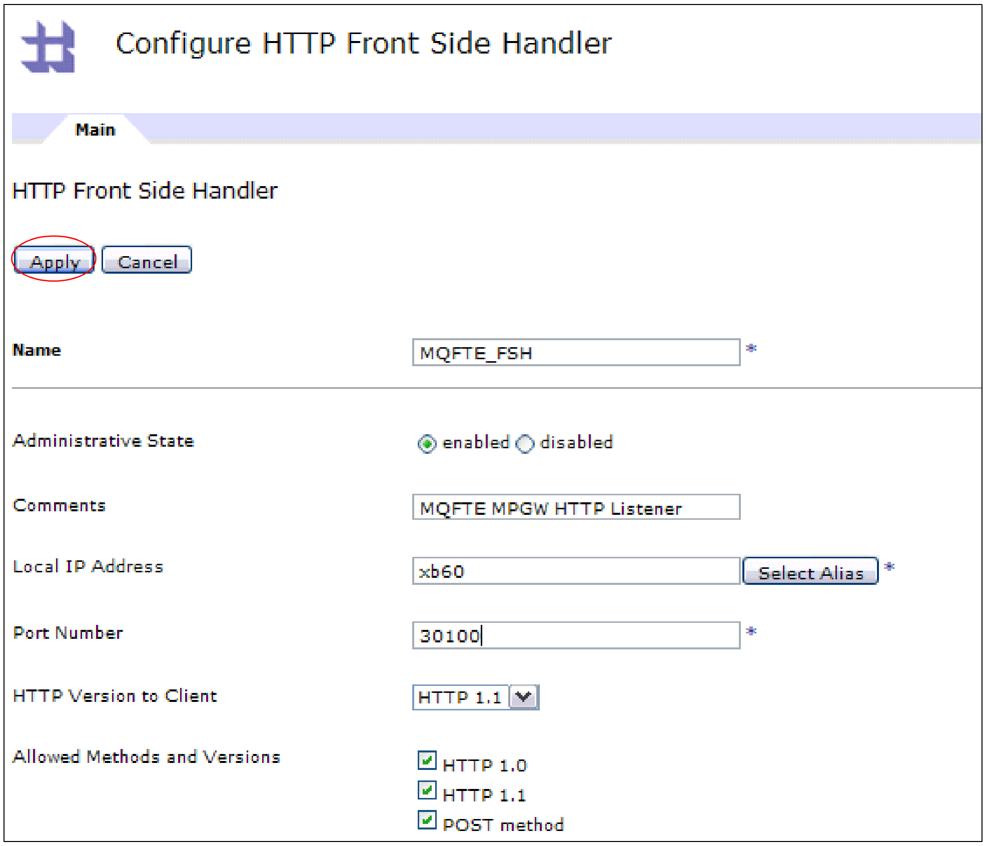


Figure 7-24 Adding a new front side protocol handler

This launches the Configure HTTP Front Side Handler view (Figure 7-25).



Configure HTTP Front Side Handler

Main

HTTP Front Side Handler

Apply | Cancel

Name	MQFTE_FSH *
Administrative State	● enabled ○ disabled
Comments	MQFTE MPGW HTTP Listener
Local IP Address	xb60 Select Alias *
Port Number	30100 *
HTTP Version to Client	HTTP 1.1
Allowed Methods and Versions	☑ HTTP 1.0 ☑ HTTP 1.1 ☑ POST method

Figure 7-25 Configure HTTP Front Side Handler

Complete the fields as follows:

i. In the Configure HTTP Front Side Handler view, on the Main tab type `MQFTE_FSH` in the Name field.

ii. Enter `xb60` in the Local IP Address field.

iii. In the Port Number field type an unused port number. For this scenario use `30100`.

iv. Leave the defaults for all other fields and click **Apply**.

v. Click **Apply** in the Configure Multi-Protocol Gateway view to finish the multi-protocol gateway configuration.

l. Save the configuration by clicking **Save Config** in the upper right corner of the view.

7.4.6 Configuring trading partner profiles

The B2B Partner Profile is the configuration object where the trading partner information is defined. This information includes the profile name, profile type, business IDs, AS security, destinations for document routing, and contact information. A trading relationship consists of, at a minimum, one internal and one external profile. For more detailed information about profile types, see the “B2B Partner Profiles” section in the *IBM WebSphere DataPower B2B Appliance XB60 Revealed*, SG24-7745.

Configuring your company's profile (internal)

To create your internal profile, do the following steps:

1. Click **B2B Partner Profile** from the Control Panel (Figure 7-26). If you are not in the Control Panel, click it on the top of the left navigation menu.

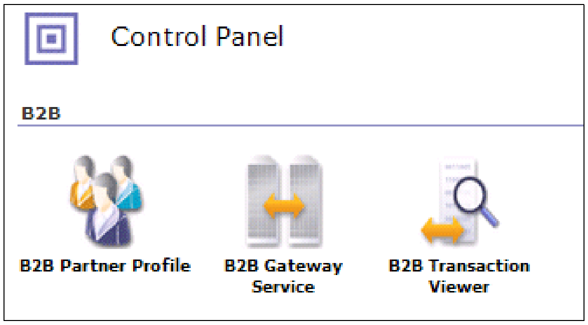
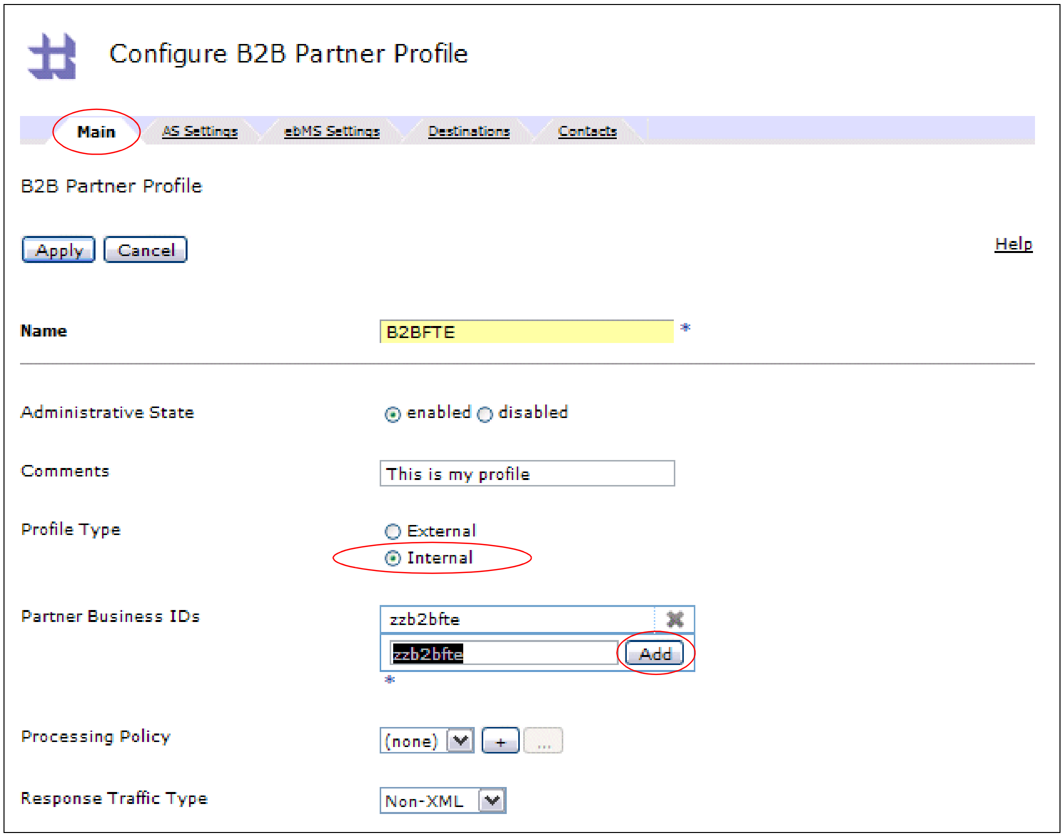


Figure 7-26 B2B portion of the control panel

2. In the Configure B2B Partner Profile list view, click **Add**. This takes you to the Configure the Partner Profile Main tab (Figure 7-27).



Configure B2B Partner Profile

B2B Partner Profile

Name: B2BFTE *

Administrative State: ● enabled ○ disabled

Comments: This is my profile

Profile Type: ○ External ● Internal

Partner Business IDs: zzb2bfte; zzb2bfte Add *

Processing Policy: (none)

Response Traffic Type: Non-XML

Figure 7-27 B2BFTE internal profile - Main tab

Configure the following fields:

a. In the Name field enter a descriptive name for your internal profile. For this scenario use `B2BFTE`.

b. Choose **enabled** in the Administrative State field.

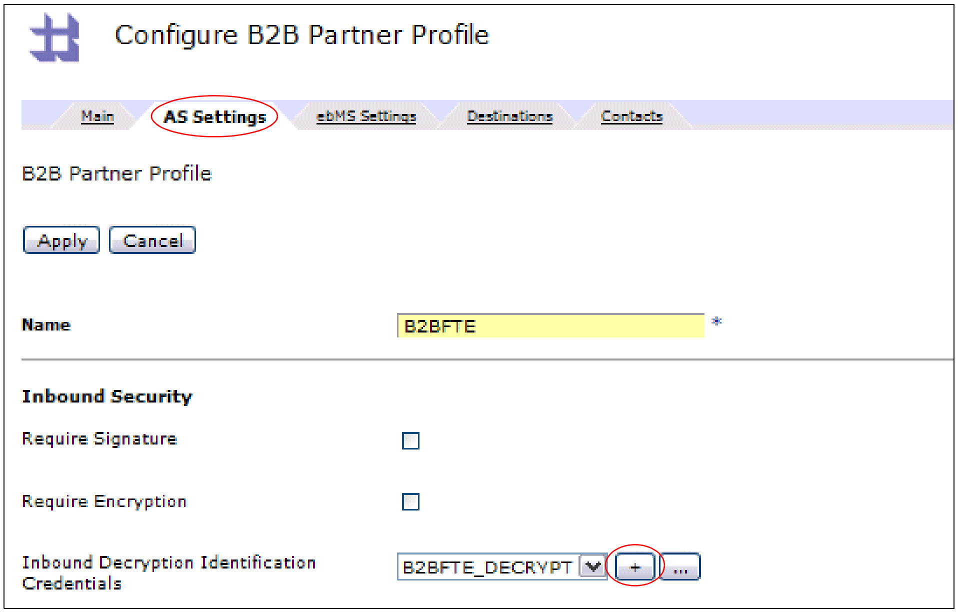
c. Choose **Internal** in the Profile Type field.

d. In the Partner Business IDs field type `zzb2bfte` and click **Add** to add the business ID to the list. Leave the default values in all other fields.

> **Important:** Do not click **Apply** at this time. The other tabs need to be configured first. Proceed to the next step.

3. Configure the partner profile AS Settings tab (Figure 7-28).

   a. Click the **AS Settings** tab. The Name field is carried over to the AS Settings panel. Do not change it.



Configure B2B Partner Profile

B2B Partner Profile

Name: B2BFTE *

Inbound Security

Require Signature □

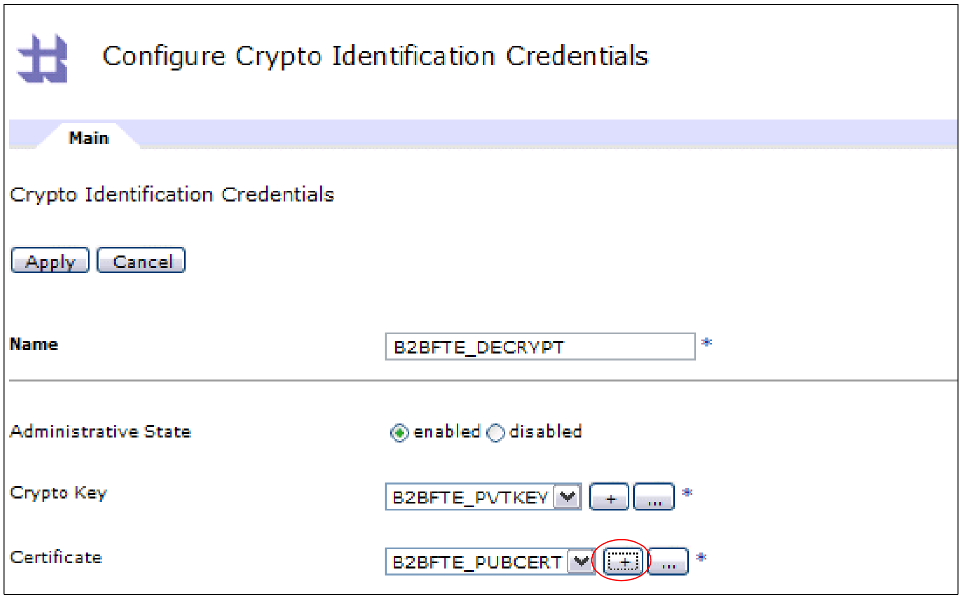
Require Encryption □

Inbound Decryption Identification Credentials: B2BFTE_DECRYPT

Figure 7-28 B2BFTE internal profile - AS Settings tab

   b. In the Inbound Security section leave the Require Signature and Require Encryption boxes unchecked.

   c. Click the Plus icon to create a new inbound decryption identification credential. This opens the panel in Figure 7-29.



Configure Crypto Identification Credentials

Crypto Identification Credentials

Name: B2BFTE_DECRYPT *

Administrative State: ● enabled ○ disabled

Crypto Key: B2BFTE_PVTKEY *

Certificate: B2BFTE_PUBCERT *

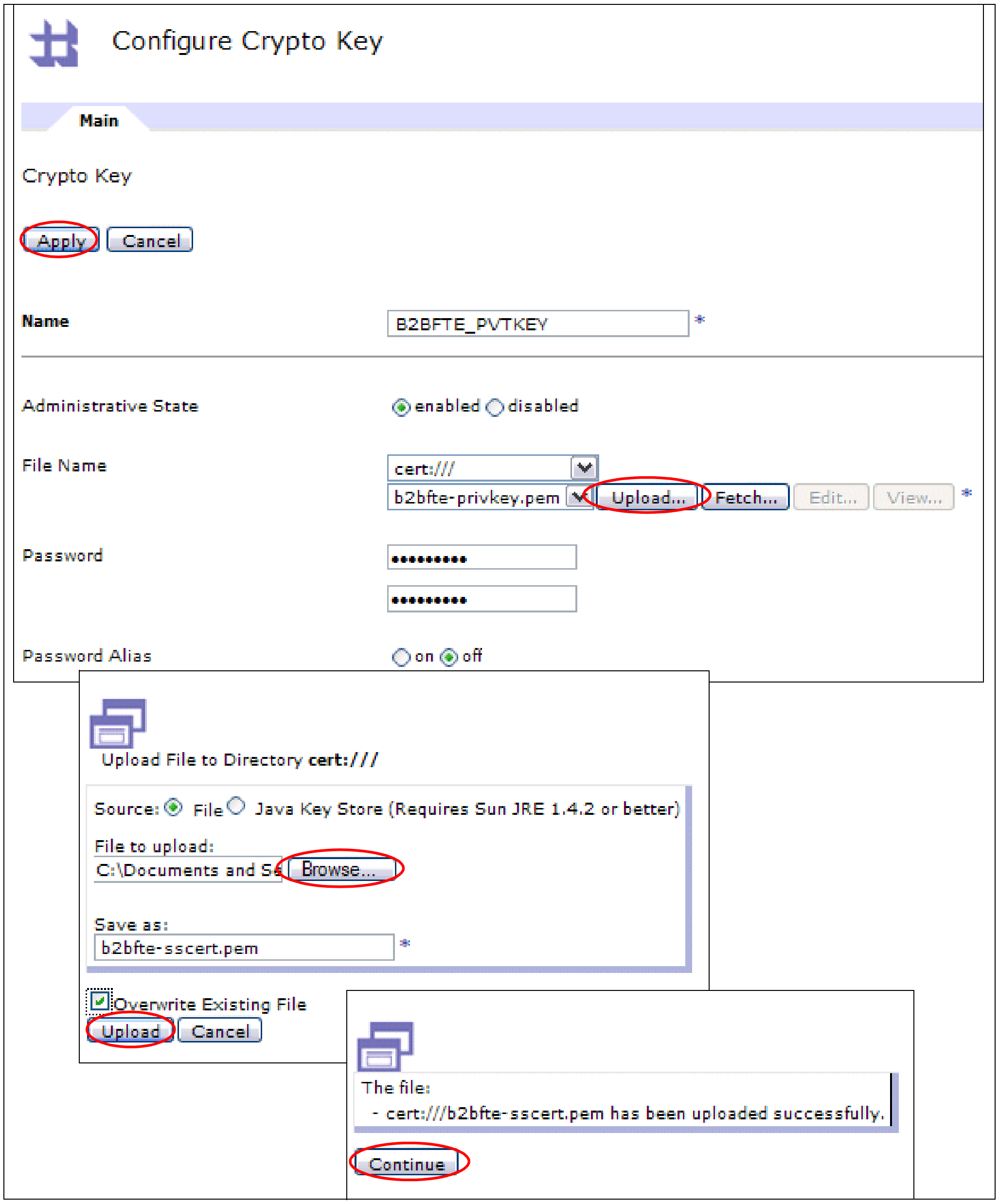
Figure 7-29 AS Settings tab - Crypto Identification Credentials

In the Configure Crypto Identification Credentials panel configure the following fields (Figure 7-30):

i. In the Name field enter a descriptive name for this credential. For this scenario use B2BFTE_DECRYPT.

ii. Choose **enabled** for the Administrative State field.

d. Click the Plus icon next to the Crypto Key field to create and upload the Crypto Key.



Configure Crypto Key

Main

Crypto Key

Name: B2BFTE_PVTKEY *

Administrative State: ● enabled ○ disabled

File Name: cert:/// b2bfte-privkey.pem *

Password: ••••••••• •••••••••

Password Alias: ○ on ● off

Upload File to Directory **cert:///**

Source: ● File ○ Java Key Store (Requires Sun JRE 1.4.2 or better)

File to upload: C:\Documents and Se

Save as: b2bfte-sscert.pem *

☑ Overwrite Existing File

The file:
- cert:///b2bfte-sscert.pem has been uploaded successfully.

Figure 7-30 AS Settings tab - inbound security crypto key

Complete the fields as follows:

i. In the Configure Crypto Key panel, in the Name field, enter a descriptive name for this key. For this scenario use B2BFTE_PVTKEY.

ii. Choose **enabled** in the Administrative State field.

iii. Click **Upload** in the File Name field.

iv. On the Upload File to Directory panel, ensure that the source is set to **File**, click **Browse** in the File to Upload field, navigate to the `b2bfte-privkey.pem` file, and then click **Open**.

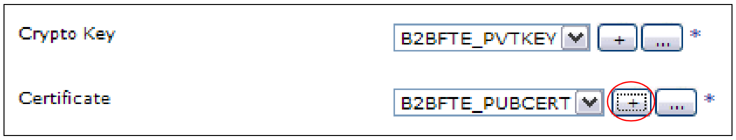
> **Additional material:** The `b2bfte-privkey.pem` file is located in the `B2BScenario_Files\certs` directory.

v. Ensure that **Overwrite Existing File** is selected, click **Upload**, and then click **Continue** in the confirmation box to return to the Configure Crypto Key view.

vi. Private keys require a password. The keys for this scenario are set to use datapower as the password. Type `datapower` in both password fields.

vii. Ensure that the Password Alias field is set to **off**, and click **Apply** to return to the Configure Crypto Identification Credentials panel.

e. Now that we are back on the Configure Crypto Identification Credentials panel (Figure 7-29 on page 141), we need to upload the certificate that is associated with the key. Click the Plus icon next to the Certificate field to create or upload the certificate (Figure 7-31).

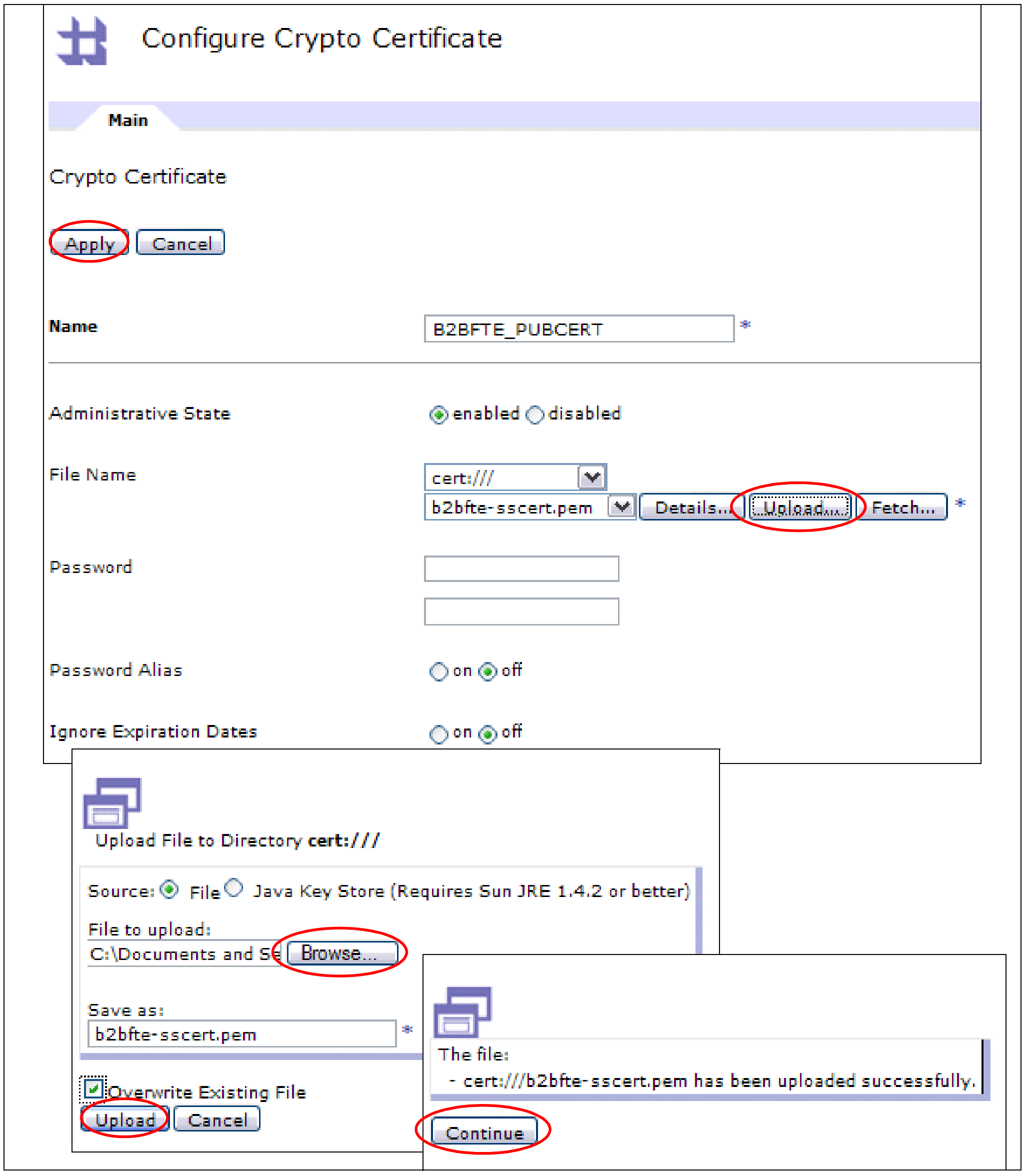


Crypto Key: B2BFTE_PVTKEY *

Certificate: B2BFTE_PUBCERT *

Figure 7-31 AS Settings tab - Crypto identification credentials - Certificate

f. The next panel is the Configure Crypto Certificate panel. Figure 7-32 shows this panel, along with the panels required to upload the certificate.



Configure Crypto Certificate

Crypto Certificate

Name: B2BFTE_PUBCERT *

Administrative State: ● enabled ○ disabled

File Name: cert:/// b2bfte-sscert.pem *

Password

Password Alias: ○ on ● off

Ignore Expiration Dates: ○ on ● off

Upload File to Directory **cert:///**

Source: ● File ○ Java Key Store (Requires Sun JRE 1.4.2 or better)

File to upload: C:\Documents and Se

Save as: b2bfte-sscert.pem *

☑ Overwrite Existing File

The file:
- cert:///b2bfte-sscert.pem has been uploaded successfully.

Figure 7-32 AS Settings tab - inbound security crypto certificate

In the Configure Crypto Certificate panel, configure the following fields:

i. In the Name field, enter a descriptive name for this certificate. For this scenario use `B2BFTE_PUBCERT`.

ii. Choose **enabled** in the Administrative State field.

iii. Click **Upload** in the File Name field. Ensure that the source is set to File.

iv. Click **Browse** in the File to Upload field, navigate to the `b2bfte-sscert.pem` file, and click **Open**.

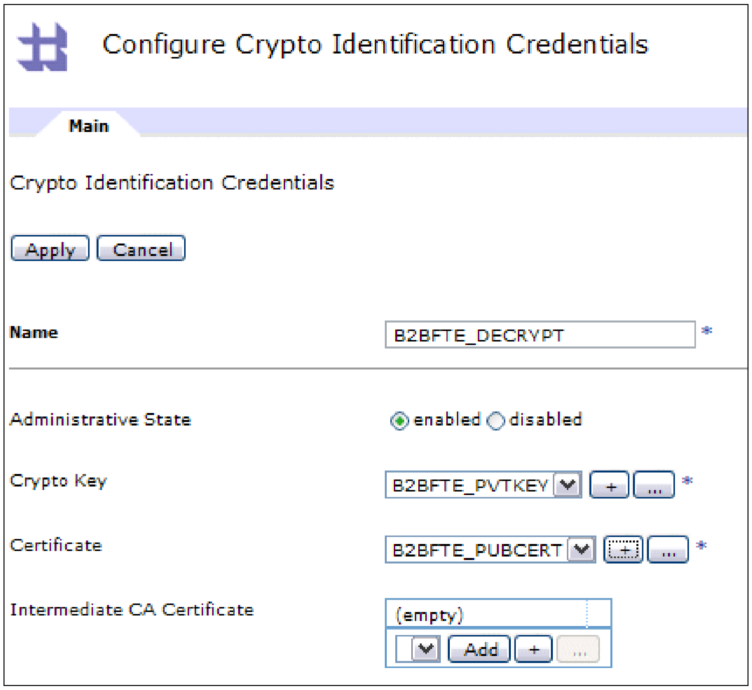
> **Additional material:** The `b2bfte-sscert.pem` file is located in the `B2BScenario_Files\certs` directory.

v. Be sure that **Overwrite Existing File** is checked, click **Upload**, and click **Continue** in the confirmation box to return to the Configure Crypto Certificate panel.

vi. In the Configure Crypto Certificate panel, leave the password fields blank. They are not needed for the public self-signed public certificate.

vii. Be sure that the Password Alias and Ignore Expiration Dates fields are set to **off**, and click **Apply** to return to the Configure Crypto Identification Credentials panel.

g. In the Configure Crypto Identification Credentials panel, leave the Intermediate CA Certificate field empty and click **Apply** to return to the AS Security tab. Figure 7-33 illustrates the completed decryption credentials configuration.



Configure Crypto Identification Credentials

Main

Crypto Identification Credentials

Apply Cancel

Name: B2BFTE_DECRYPT *

Administrative State: ● enabled ○ disabled

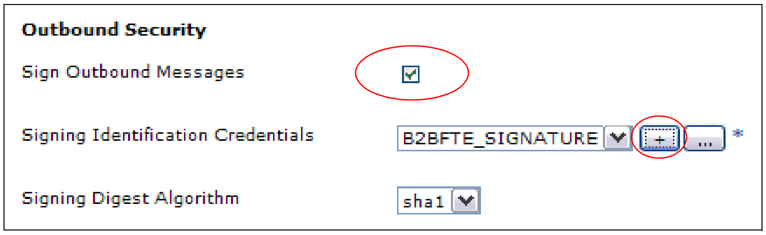
Crypto Key: B2BFTE_PVTKEY *

Certificate: B2BFTE_PUBCERT *

Intermediate CA Certificate: (empty)

Figure 7-33 AS Settings tab - Crypto Identification Credentials - Finished

h. In the AS Settings view, in the Outbound Security section, be sure that the **Sign Outbound Messages** box is checked and click the Plus icon to create a new Outbound Signing Identification Credential (Figure 7-34).



Outbound Security

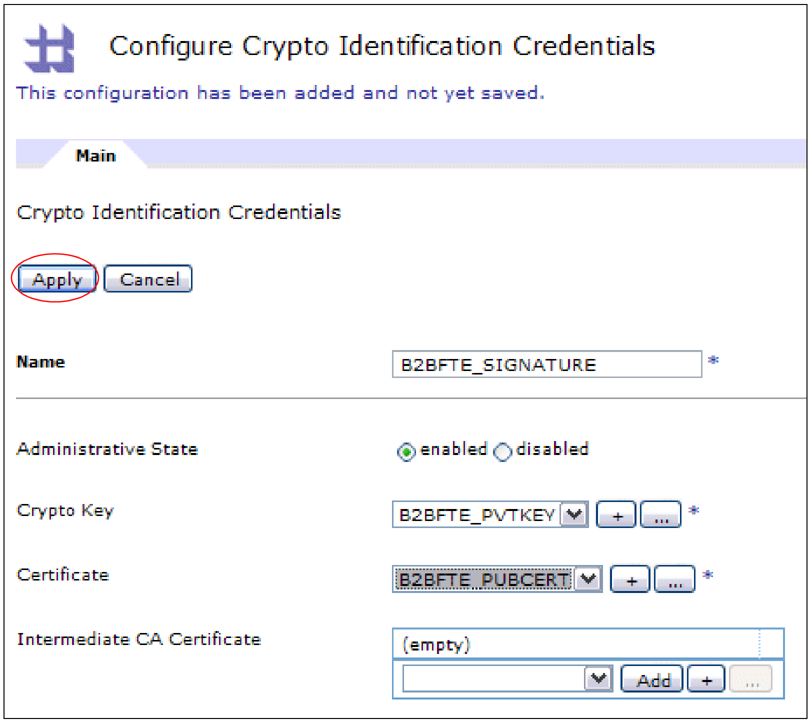
Sign Outbound Messages: ☑

Signing Identification Credentials: B2BFTE_SIGNATURE *

Signing Digest Algorithm: sha1

Figure 7-34 B2BFTE internal profile - AS Settings tab - Step C

This opens the Configure Crypto Identification Credentials panel (Figure 7-35).



Configure Crypto Identification Credentials

This configuration has been added and not yet saved.

Crypto Identification Credentials

Name: B2BFTE_SIGNATURE *

Administrative State: ● enabled ○ disabled

Crypto Key: B2BFTE_PVTKEY *

Certificate: B2BFTE_PUBCERT *

Intermediate CA Certificate: (empty)

Figure 7-35 B2BFTE internal profile - AS Settings tab - Outbound security

Configure the following fields:

i. In the Name field enter a descriptive name for this credential. For this use `B2BFTE_SIGNATURE` (Figure 7-35).

ii. Choose **enabled** in the Administrative State field.

iii. We use the same certificates for signing and decryption. Because we already imported the key and certificate for the decryption credentials, click the drop-down menu and select **B2BFTE_PVTKEY** for the Crypto Key and **B2BFTE_PUBCERT** for the certificate.

iv. Leave the Intermediate CA Certificate field empty because we are using self-signed certificates.

v. After both credentials are configured click **Apply**, which puts you back into the AS Settings view (Figure 7-34 on page 145).

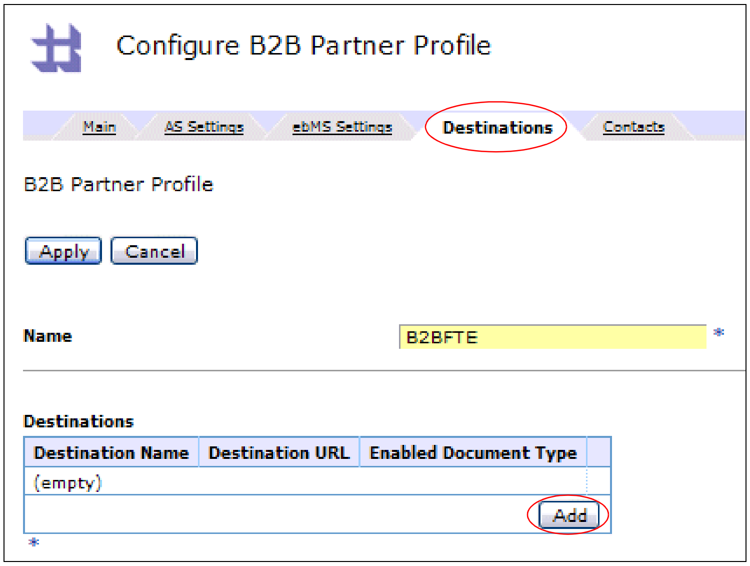
i. With the addition of the Signing Identification Credential, a Signing Digest Algorithm field now appears under the Credential field. Leave this set to sha1 and leave all remaining fields set to their default values.

> **Important:** Do not click Apply at this time. The other tabs need to be configured first. Proceed to the next step.

4. Because we are not using the ebMS protocol, we do not need to configure the ebMS Settings tab of the B2B partner profile.

5. Configure the Destinations tab:

   a. Click the **Destinations** tab (Figure 7-36). The Name field will carry over to the Destinations view. Do not change it.



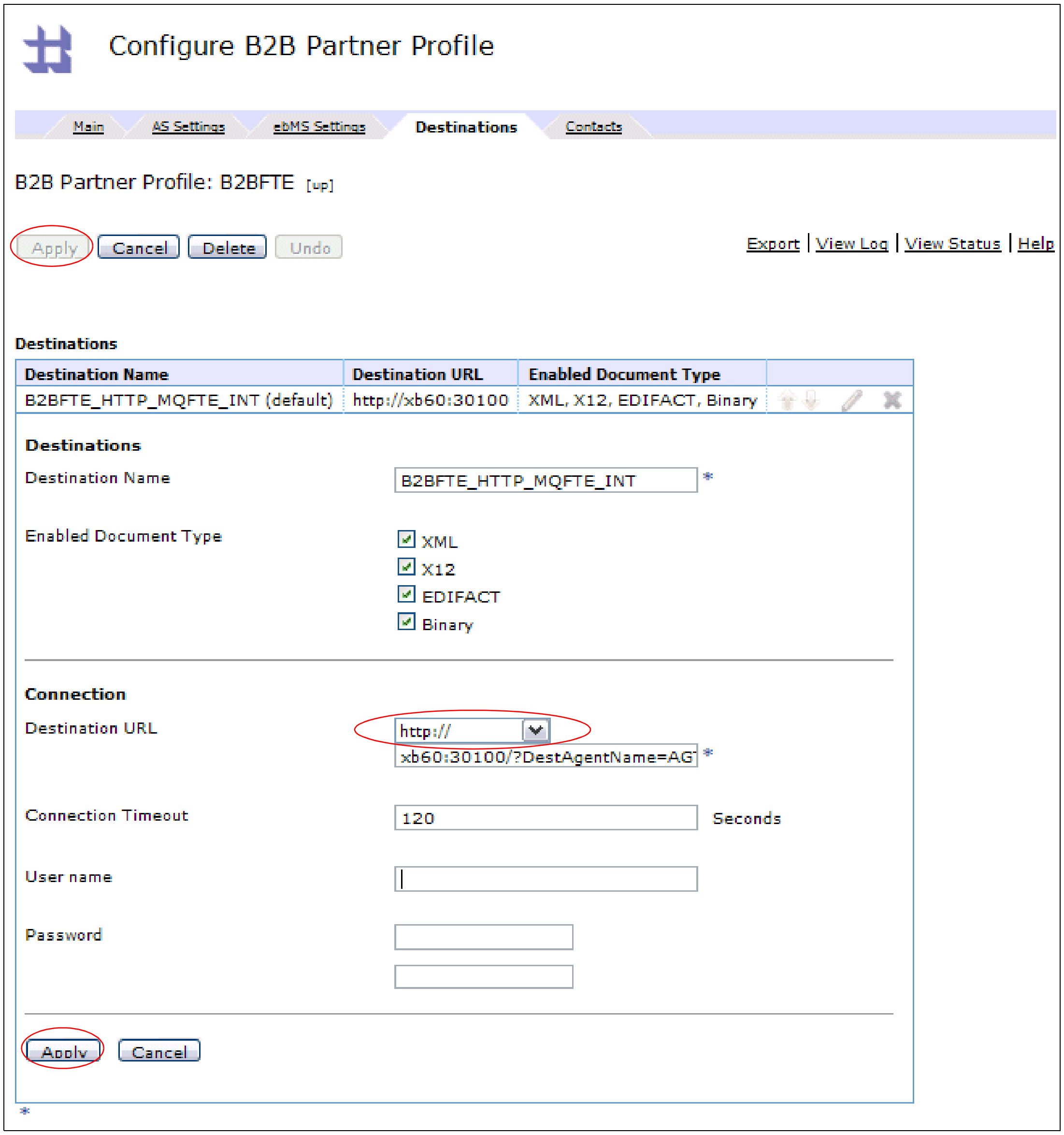
Name: B2BFTE *

Destinations

Destination Name	Destination URL	Enabled Document Type
(empty)		

Figure 7-36 B2BFTE internal profile - Add destination

b. In the Destinations view click **Add** to add a destination to this profile. Because this profile is an internal profile, the destination typically will be a system or application inside your private network. For the purpose of this scenario, use HTTP as the destination. This is the handler/listener that you created in the multi-protocol gateway that you created earlier in this chapter (Figure 7-37).



Configure B2B Partner Profile

B2B Partner Profile: B2BFTE [up]

Destinations

Destination Name	Destination URL	Enabled Document Type
B2BFTE_HTTP_MQFTE_INT (default)	http://xb60:30100	XML, X12, EDIFACT, Binary

Destinations

Destination Name: B2BFTE_HTTP_MQFTE_INT *

Enabled Document Type:
☑ XML
☑ X12
☑ EDIFACT
☑ Binary

Connection

Destination URL: http:// xb60:30100/?DestAgentName=AG *

Connection Timeout: 120 Seconds

User name:

Password:

Figure 7-37 B2BFTE internal profile - HTTP destination

Configure the following fields:

i. Enter a descriptive name in the Destination Name field. For the purpose of this scenario use `B2BFTE_HTTP_MQFTE_INT`.

ii. Leave all of the boxes checked in the Enable Document Type section. This allows your internal profile to accept and produce all supported file types.

iii. In the Connection section use the drop-down menu to select **http://** as the Destination URL and type and use `xb60:30100/?DestAgentName=AGTSAFE&DestAgentQMgr=QMSAFE` as the address.

Troubleshooting tip: Information to the right of the question mark (?) is used as input to the XSL used in the MQ FTE Proxy Multi-Protocol gateway. Be sure to type the URL exactly as seen in this step, as it is case sensitive. If the file is not moved from the input directory (`/XB60/b2bfte/In`) to the destination directory (`C:\B2BFTE\In`), the problem is usually that the URL is incorrect.

- iv. Change the connection timeout to **120** seconds.
- v. Leave the User Name and Password fields blank because we are not using basic authentication.
- vi. Click **Apply** inside the Destination configuration view to return to the destination list.

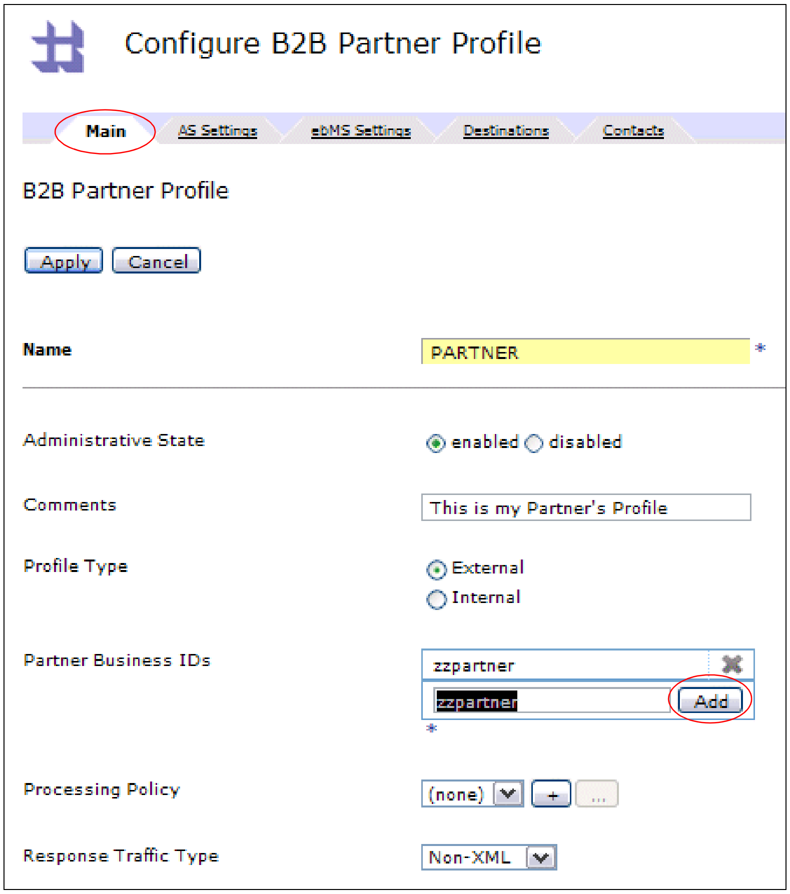
6. Configuring the Partner Profile Contacts tab is optional. We do not configure contacts for this scenario.
7. Now that the internal profile is completely configured click **Apply** for the profile.
8. Save the profile by clicking **Save Config** in the upper right corner of the web page.

Configure your partner's profile (external)

Configuring your external partner's profile is done in the same manner in which you configured the internal partner profile. This section does not revisit the steps at the same level of detail as done in the internal partner section. However, certain steps and changes to the required fields are presented here for you to use in the creation of the profile.

1. After creation of the internal profile you should be back in the B2B Partner Profile list view. If not, expand **Services** in the left navigation menu and click **B2B Partner Profile**.
2. Click **Add** to configure the external profile.

3. Configure the Partner Profile Main tab. Figure 7-38 illustrates how the configured external partner Main tab should look. Fields that are populated with default values might not be displayed.



Configure B2B Partner Profile

B2B Partner Profile

Name: PARTNER *

Administrative State: ● enabled ○ disabled

Comments: This is my Partner's Profile

Profile Type: ● External ○ Internal

Partner Business IDs: zzpartner; zzpartner [Add] *

Processing Policy: (none)

Response Traffic Type: Non-XML

Figure 7-38 B2BFTE external profile - Main tab

Complete these fields as follows:

a. In the Name field enter a descriptive name your partner's external profile. For this scenario use `PARTNER`.

b. Choose **enabled** in the Administrative State field.

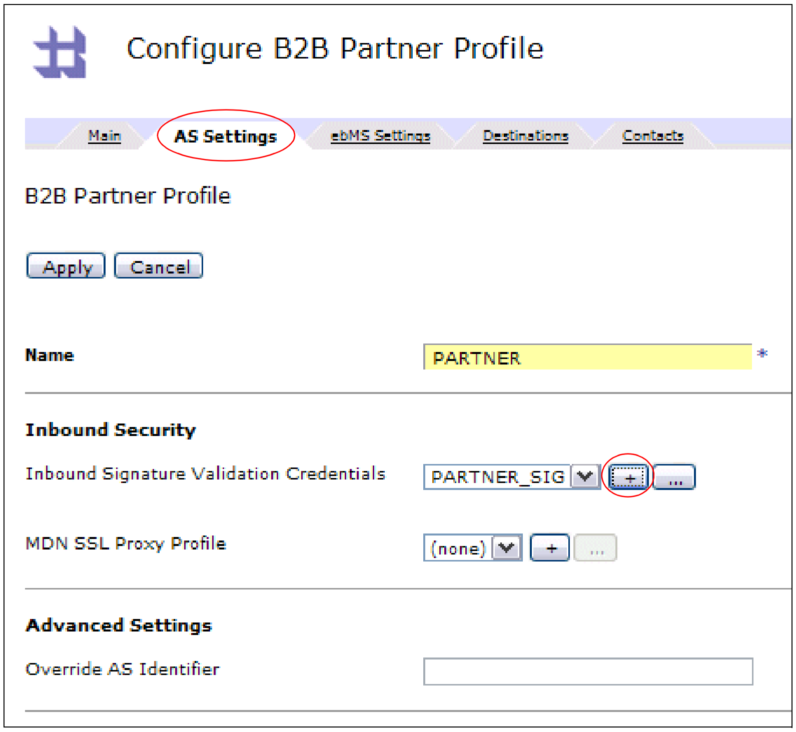
c. Optional: Add comments that describe this profile.

d. Choose **External** in the Profile Type field.

e. In the Partner Business IDs field enter your ID. For this scenario type `zzpartner` and click **Add** to add each business ID to the list. Leave the defaults in all remaining fields.

> **Important:** Do not click Apply at this time. The other tabs need to be configured first. Proceed to the next step.

f. Configure the partner profile by clicking the **AS Settings** tab. This places you in the view used to configure AS security (Figure 7-39).



Configure B2B Partner Profile

B2B Partner Profile

Name: PARTNER *

Inbound Security

Inbound Signature Validation Credentials: PARTNER_SIG

MDN SSL Proxy Profile: (none)

Advanced Settings

Override AS Identifier:

Figure 7-39 B2BFTE external profile - AS Settings tab

To create the configuration that you see here, take the following steps:

a. The Name field carries over to the AS Settings panel. Do not change it.

b. In the Inbound Security section click the Plus icon to create a new inbound signature validation credential.

   i. In the Configure Crypto Validation Credentials panel, in the Name field enter a descriptive name for this credential. For this scenario use `PARTNER_SIG`.

   ii. Choose **enabled** in the Administrative State field.

c. Click the Plus icon next to the Certificates field to create or upload the partner certificate.

   i. In the Configure Crypto Certificate panel, in the Name field, enter a descriptive name for this certificate. For this scenario use `PARTNER_CERT`.

   ii. Choose **enabled** in the Administrative State field.

   iii. Click **Upload** in the File Name field.

d. In the Upload File panel be sure that the source is File, click **Browse** in the File to Upload field, navigate to the `partner-sscert.pem` file, and click **Open**.

> **Additional material:** The `partner-sscert.pem` file is located in the `B2BScenario_Files\certs` directory.

Be sure that Overwrite Existing File is checked, click **Upload**, and click **Continue** in the upload success box. This take you back to the Configure Crypto Certificate panel.

e. Leave the password fields blank. This is a public certificate, so there is no need for a password.

   Be sure that the Password Alias and Ignore Expiration Date fields are set to off and click **Apply**. This take you back to the Configure Crypto Validation Credentials panel.

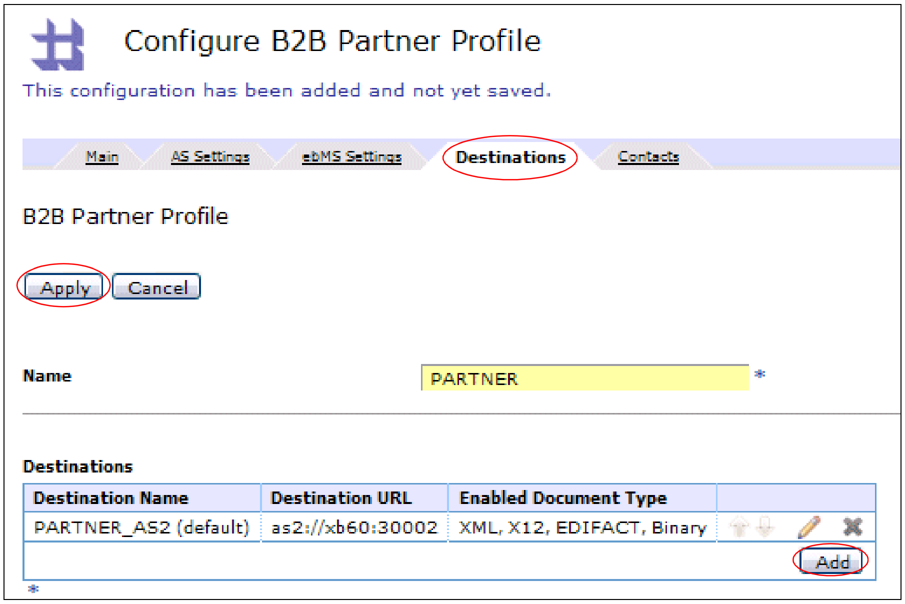
f. In the Configure Crypto Validation Credentials panel accept the default value, Match exact certificate or immediate issuer, in the Certificate Validation Mode field.

   Be sure that the Use CRL field is set to off and click **Apply**. This take you back to the AS Settings panel.

g. On the AS Settings tab (Figure 7-39 on page 151), set the MDN SSL Proxy Profile to **(none)** and leave the default values for all the remaining fields.

> **Important:** Do not click Apply at this time. The other tabs need to be configured first. Proceed to the next step.

4. Configure the Partner Profile Destinations tab as shown in Figure 7-40.



Configure B2B Partner Profile

This configuration has been added and not yet saved.

B2B Partner Profile

Name: PARTNER *

Destinations

Destination Name	Destination URL	Enabled Document Type	
PARTNER_AS2 (default)	as2://xb60:30002	XML, X12, EDIFACT, Binary	

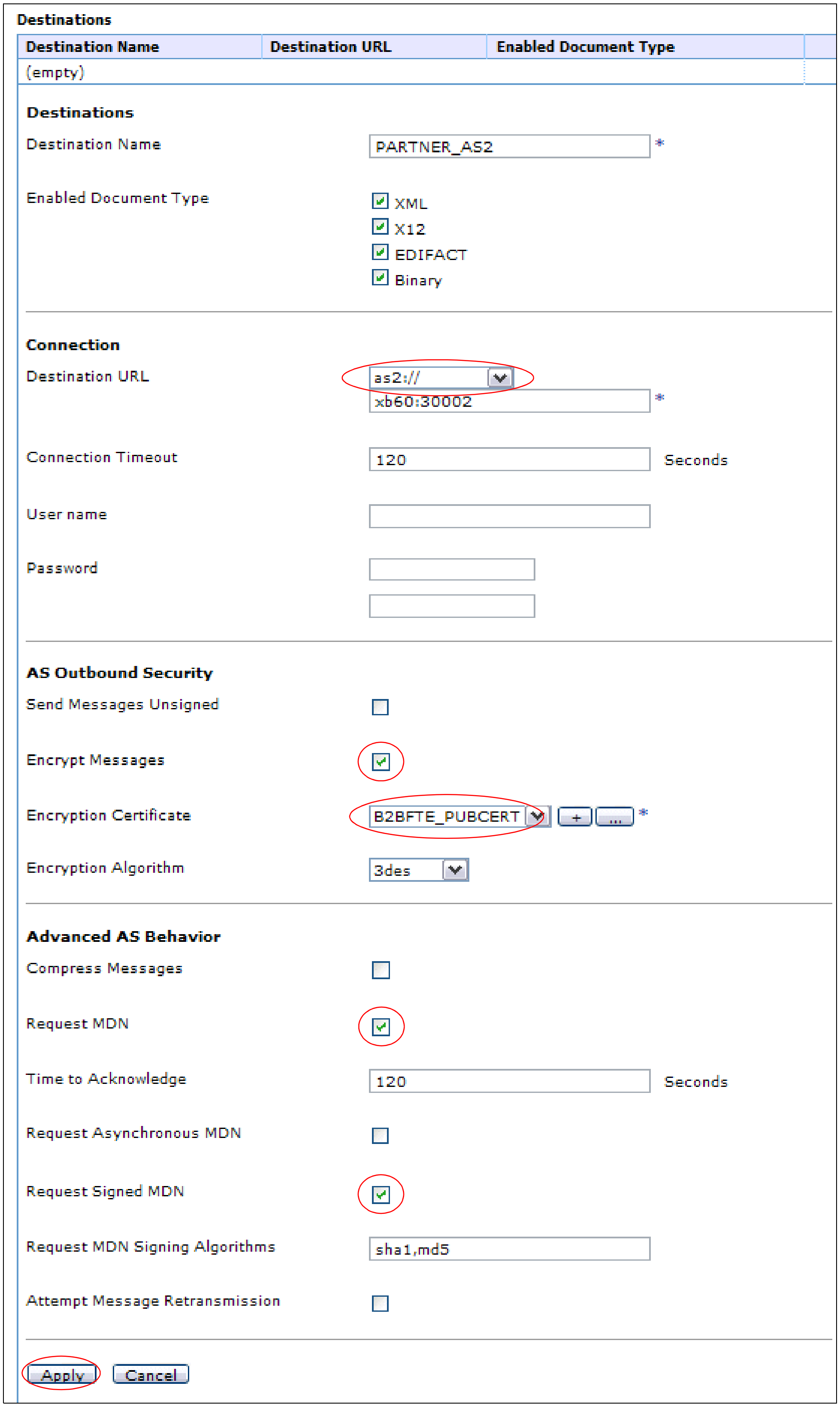
Figure 7-40 B2BFTE external profile - Destinations tab

Complete these fields as follows:

a. The Name field carries over into the Destinations tab. Do not change it.

b. In the Destinations section click **Add** to add a destination to this profile. For this scenario we use an AS2 destination.

Figure 7-41 shows the configured destination.



Destinations

Destination Name	Destination URL	Enabled Document Type
(empty)		

Destinations

Destination Name: PARTNER_AS2 *

Enabled Document Type: ☑ XML ☑ X12 ☑ EDIFACT ☑ Binary

Connection

Destination URL: as2:// xb60:30002 *

Connection Timeout: 120 Seconds

User name:

Password:

AS Outbound Security

Send Messages Unsigned: □

Encrypt Messages: ☑

Encryption Certificate: B2BFTE_PUBCERT *

Encryption Algorithm: 3des

Advanced AS Behavior

Compress Messages: □

Request MDN: ☑

Time to Acknowledge: 120 Seconds

Request Asynchronous MDN: □

Request Signed MDN: ☑

Request MDN Signing Algorithms: sha1,md5

Attempt Message Retransmission: □

Figure 7-41 B2BFTE external profile - AS2 destination configuration

To create this configuration:

i. Enter a descriptive name in the Destination Name field. For this scenario use `PARTNER _AS2`.

ii. Leave all of the boxes checked in the Enable Document Type section. This allows this profile to accept and produce all supported file types.

iii. In the Connection section, use the drop-down menu to select **as2://** as the Destination URL Type and type `xb60:30002` as the address. This displays the AS2 Destination configuration view as seen in Figure 7-41 on page 153.

iv. Change the connection timeout to **120** seconds.

v. Leave the User Name and Password fields blank because we are not using basic authentication.

vi. In the AS Outbound Security section, leave the Send Messages Unsigned box unchecked. We want messages to be signed based on the settings in the AS security configuration in the sending partner's profile.

vii. Place a check in the box next to the Encrypt Messages field. An Encryption Certificate field is now visible. This allows us to encrypt the payload data that we send to the partner.

viii.In the Encryption Certificate field use the drop-down menu and select the same public certificate credential that we created for validating signatures, **PARTNER_CERT**, and leave the encryption algorithm as 3des.

ix. In the Advanced AS Behavior section, leave the Compress Messages box unchecked.

x. Place a check in the box next to the Request MDN field. This makes additional MDN fields visible, allowing us to request an MDN back from the partner verifying that they received the file.

xi. Set the Time to Acknowledge field to **120**. Because we use sync MDNs, we want the timeout to be smaller than the default of 30 minutes (1800 ms).

xii. Place a check in the box next to the Request Signed MDN field. This makes the Request MDN Signing Algorithms field available. These settings allow us to request that the returned MDN be signed.

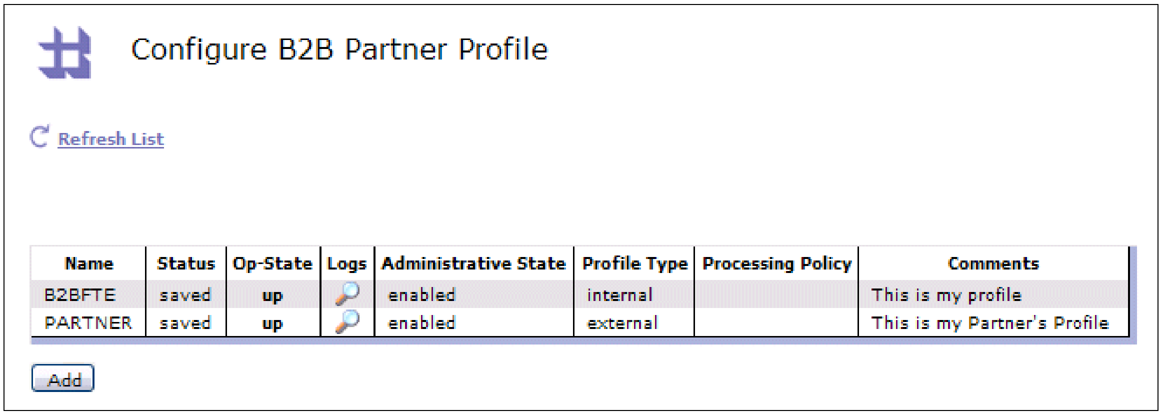
xiii.Leave the Request MDN Signing Algorithms field set to sha1,md5.

xiv.Leave the Attempt Message Retransmission box unchecked. This setting is used if you want to configure automatic resends to be used in the event that an MDN is not received.

xv. Click **Apply** in the Destination box to return to the Destination List. You might need to scroll back to the top of the page after creating the destination to see the configured fields.

5. Now that the external profile is completely configured, click **Apply** for the profile.
6. Click **Save Config** in the upper right of the web page to persist the changes.

When both profiles are added correctly you will see a profile list view like Figure 7-42.



Configure B2B Partner Profile

Refresh List

Name	Status	Op-State	Logs	Administrative State	Profile Type	Processing Policy	Comments
B2BFTE	saved	up	🔎	enabled	internal		This is my profile
PARTNER	saved	up	🔎	enabled	external		This is my Partner's Profile

Figure 7-42   Partner profiles list view - Finished

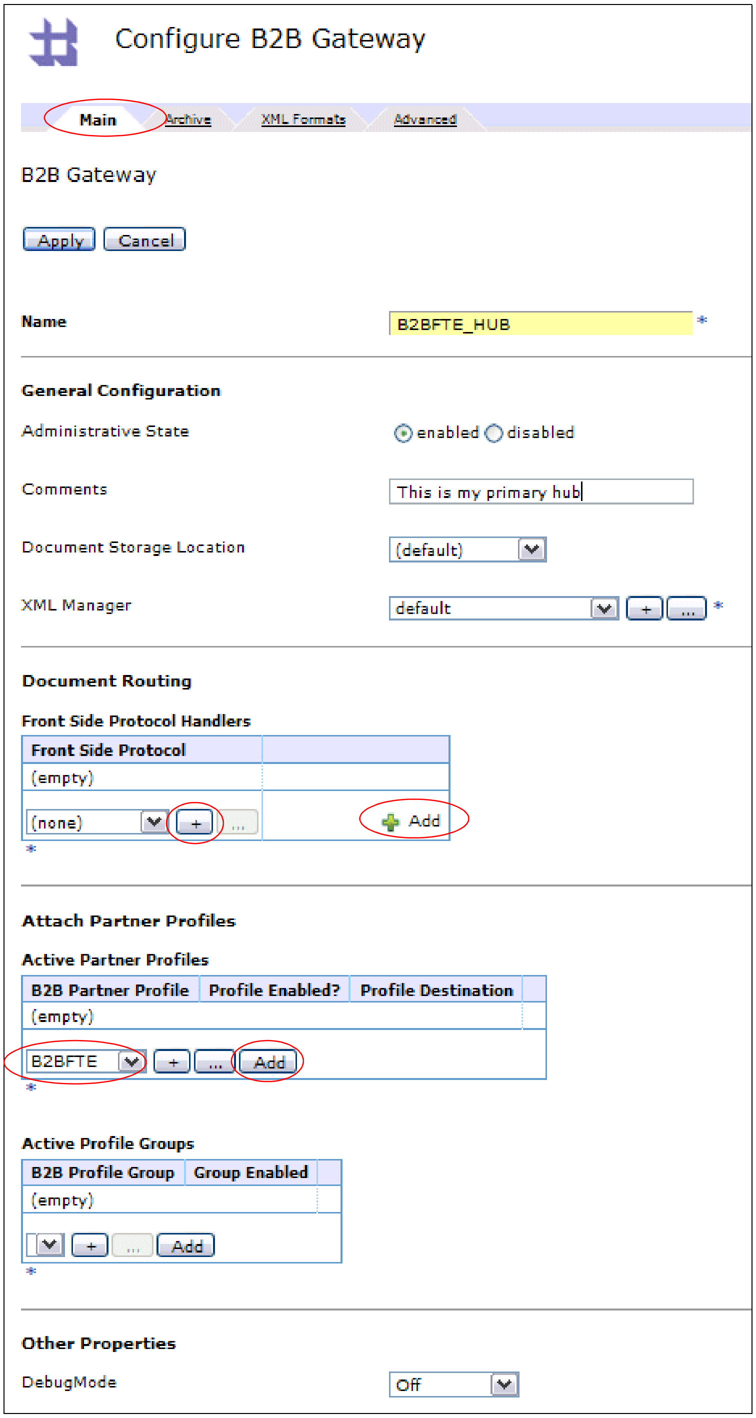
7.4.7 Configuring the B2B gateway

Your B2B gateway is the primary B2B hub and is depicted in the scenarios as the owner of the XB60. In this section you are configuring the B2BFTE_HUB B2B gateway to trade with a single partner's B2B hub. The partner's B2B hub can be any AS2 interoperable product. However, for the purpose of this exercise we simulated the partner's B2B hub in a separate domain on the same XB60 being used for this scenario.

To configure the B2B gateway:

1. Expand **Services** in the left navigation menu of the XB60 and click **B2B Gateway Service**.
2. In the Configure B2B Gateway list view, click **Add**.

3. Configure the B2B Gateway Main tab as shown in Figure 7-43.



B2B Gateway

Name: B2BFTE_HUB *

General Configuration

Administrative State: ● enabled ○ disabled

Comments: This is my primary hub

Document Storage Location: (default)

XML Manager: default *

Document Routing

Front Side Protocol Handlers

Front Side Protocol	
(empty)	
(none)	Add

*

Attach Partner Profiles

Active Partner Profiles

B2B Partner Profile	Profile Enabled?	Profile Destination
(empty)		
B2BFTE		

*

Active Profile Groups

B2B Profile Group	Group Enabled
(empty)	

*

Other Properties

DebugMode: Off

Figure 7-43 B2BFTE_HUB Gateway - Main tab

Complete these fields as follows:

a. Enter the B2B gateway name in the Name field. For the purpose of this scenario use `B2BFTE_HUB`.

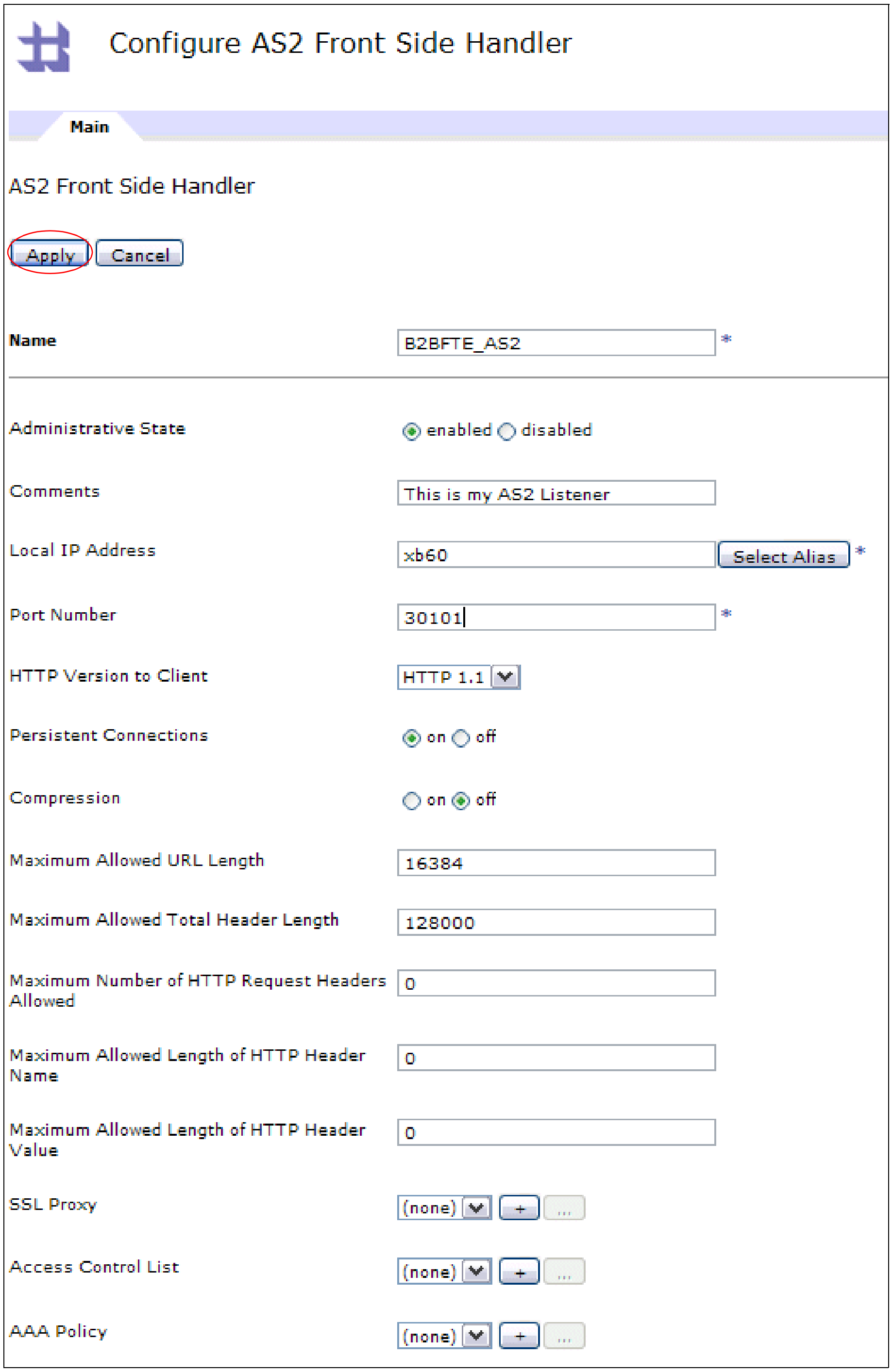
b. Choose **enabled** in the Administrative State field.

c. Optional: Add comments that describe this gateway.

d. Accept the defaults for the Document Storage Location and XML Manager fields.

e. In the Document Routing section create and configure an AS2 front side handler. This handler is used to receive AS2 messages and MDNs from the trading partner.

   i. Click the Plus icon to create a new handler and choose **AS2 Front Side Handler** from the list of handlers. This launches the Configure AS2 Front Side Handler view. Figure 7-44 illustrates how the configured AS2 front side handler should look. Fields that are populated with default values might not be displayed.



Configure AS2 Front Side Handler

AS2 Front Side Handler

Name	B2BFTE_AS2 *
Administrative State	● enabled ○ disabled
Comments	This is my AS2 Listener
Local IP Address	xb60 [Select Alias] *
Port Number	30101 *
HTTP Version to Client	HTTP 1.1
Persistent Connections	● on ○ off
Compression	○ on ● off
Maximum Allowed URL Length	16384
Maximum Allowed Total Header Length	128000
Maximum Number of HTTP Request Headers Allowed	0
Maximum Allowed Length of HTTP Header Name	0
Maximum Allowed Length of HTTP Header Value	0
SSL Proxy	(none) [+] [...]
Access Control List	(none) [+] [...]
AAA Policy	(none) [+] [...]

Figure 7-44 B2BFTE_HUB Gateway - AS2 front side protocol handler

ii. In the Main tab, for the Name field enter `B2BFTE_AS2`.

iii. Choose **enabled** in the Administrative State field.

iv. Optional: Add comments that describe this handler.

v. Type `xb60` in the Local IP Address field.

vi. In the Port Number field enter `30101`.

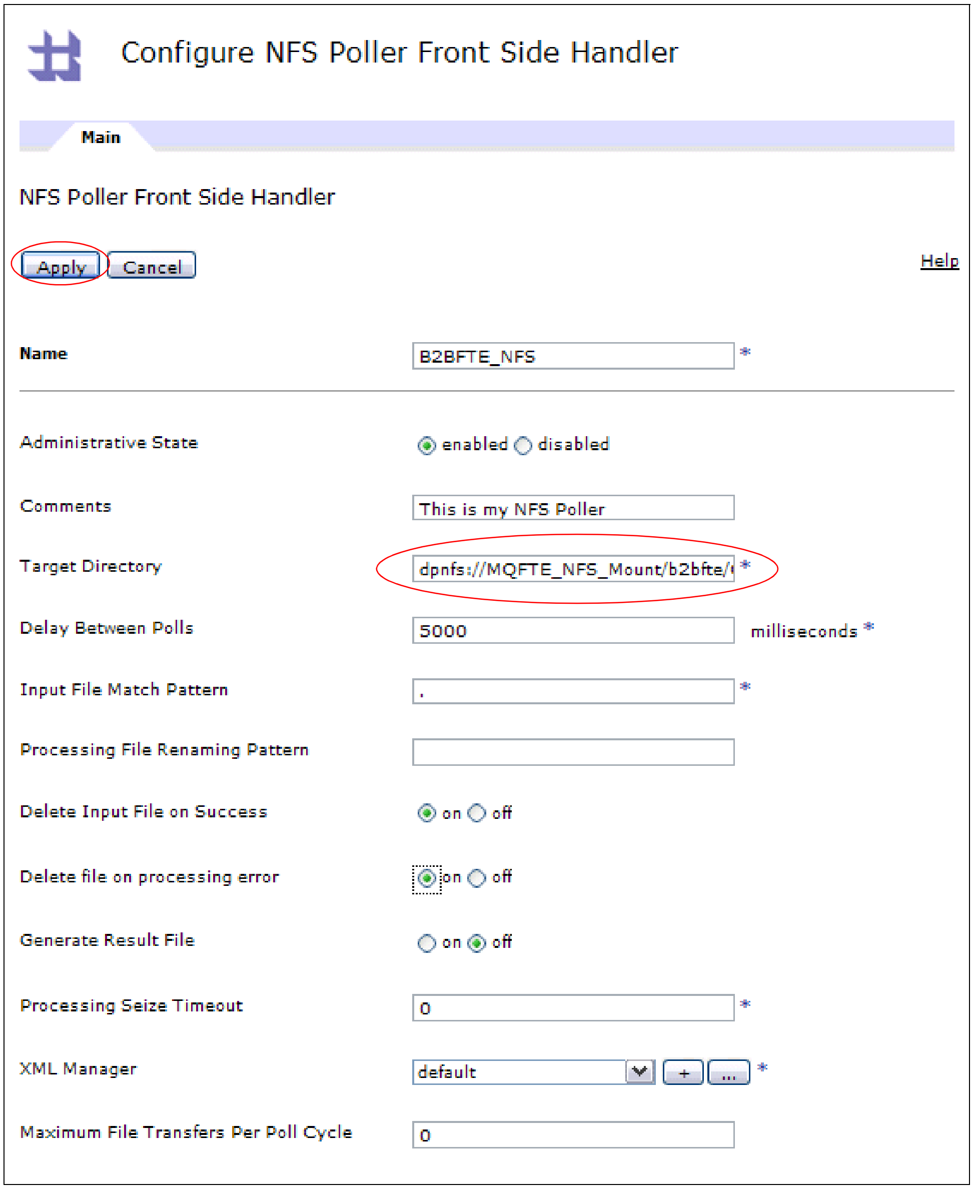
vii. Take the default values for all other fields and click **Apply**.

viii.Click the **+ Add** link to add the listener to the Front Side Protocol list.

> **Troubleshooting tip:** The AS2 front side handler is an http listener where external partner's are sending AS2 messages to you. Optionally, you can use SSL with the front side handler if you want to provide connection security in addition to the data security that AS2 already offers. The steps for using SSL with AS2 are not documented in this book. Refer to the Xb60 user documentation for instructions on how to configure an SSL Proxy. The port number is any port that you want the AS2 handler to listen on. Be sure that your partner has access to the port from the internet or that the external firewall is routing data to the port being used.

f. In the Document Routing section create and configure an NFS Poller Front Side Handler. This handler is used to receive files from the NFS file share that is associated with MQ FTE.

   i. Click the Plus icon to create a new handler. This launches a new view (Figure 7-45).



Configure NFS Poller Front Side Handler

NFS Poller Front Side Handler

Name	B2BFTE_NFS *
Administrative State	● enabled ○ disabled
Comments	This is my NFS Poller
Target Directory	dpnfs://MQFTE_NFS_Mount/b2bfte/ *
Delay Between Polls	5000 milliseconds *
Input File Match Pattern	. *
Processing File Renaming Pattern	
Delete Input File on Success	● on ○ off
Delete file on processing error	● on ○ off
Generate Result File	○ on ● off
Processing Seize Timeout	0 *
XML Manager	default *
Maximum File Transfers Per Poll Cycle	0

Figure 7-45 B2BFTE_HUB Gateway - NFS Poller Front Side Protocol Handler

   ii. Choose **NFS Poller Front Side Handler** from the list of handlers.

   iii. On the Main tab, in the Name field enter `B2BFTE_NFS`.

   iv. Choose **enabled** in the Administrative State field.

   v. Optional: Add comments that describe this handler.

   vi. In the Target Directory field enter `dpnfs://MQFTE_NFS_Mount/b2bfte/Out/`.

   vii. In the Delay Between Polls field type `5000`.

   viii.In the Input File Match Pattern field type a period (`.`) to match all.

ix. Make sure that the Delete Input File on Success field is set to on. Files must be deleted after they are picked up, otherwise they are picked up again during the next poll cycle.

x. Make sure that the Delete File on Processing Error field is set to on. It is common for the file to error because the partner's hub is down. We want these failed transfers to be resent to the partner from the gateway, not kept in the directory.

xi. Make sure that the Generate Results File field is set to off.

xii. Accept the default values for all other fields and click **Apply**.

xiii.Click the **+ Add** link to add the poller to the Front Side Protocol list.

> **Troubleshooting tip:** The NFS Poller Front Side Handler is a poller that is polling the NFS directory that is shared with WebSphere MQ File Transfer Edition. It is looking for files to send to the external partner's AS2 Listener. If your NFS mount point is inside the protected network you need to be sure that the inner firewall has a rule that allows the XB60 to send data over the NFS ports being used (typically 2049 and 111). Your internal network security policies govern where the NFS share is located and how it is protected. If files are not being picked up, your target directory value might be wrong or AGTSAFE might be down and is not sending files to `/XB60/b2bfte/Out`.

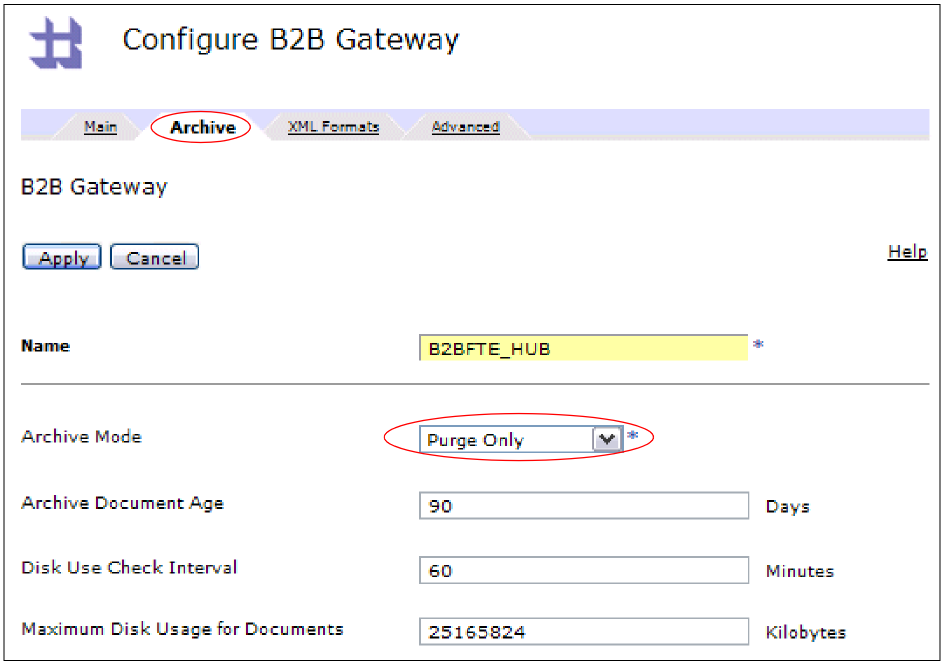
xiv.In the Attach Partner Profiles section click the drop-down menu, select **B2BFTE**, and click **Add**. Click the drop-down menu again, select **PARTNER**, and click **Add**. This associates the profiles that you created in 7.4.6, "Configuring trading partner profiles" on page 139 with your B2B gateway.

g. Skip the Active Profile Groups section. We do not use them for this scenario.

> **Important:** Do not click Apply at this time. The other tabs need to be configured first. Proceed to the next step.

4. Configure the Archive tab as shown in Figure 7-46.

> **Troubleshooting tip:** The Archive tab is used to automatically keep the B2B document and metadata storage areas clean. There are two modes:
>
> - Archive and Purge
> - Purge Only
>
> Perform capacity planning to determine how much drive space you need to support your retention policies. In certain cases, you will need more space than is available on the DataPower device's hard drive and you will need to store B2B payloads off device using one of the external hard drive storage options. See the XB60 user documentation for details on using external drives with the appliance.


Configure B2B Gateway

B2B Gateway

Name: B2BFTE_HUB *

Archive Mode: Purge Only *

Archive Document Age: 90 Days

Disk Use Check Interval: 60 Minutes

Maximum Disk Usage for Documents: 25165824 Kilobytes

Figure 7-46 B2BFTE_HUB Gateway - Archive tab

Complete these fields as follows:

a. The Name field carries over to the Archive panel. Do not change it.

b. In the Archive Mode field use the drop-down menu and select **Purge Only**.

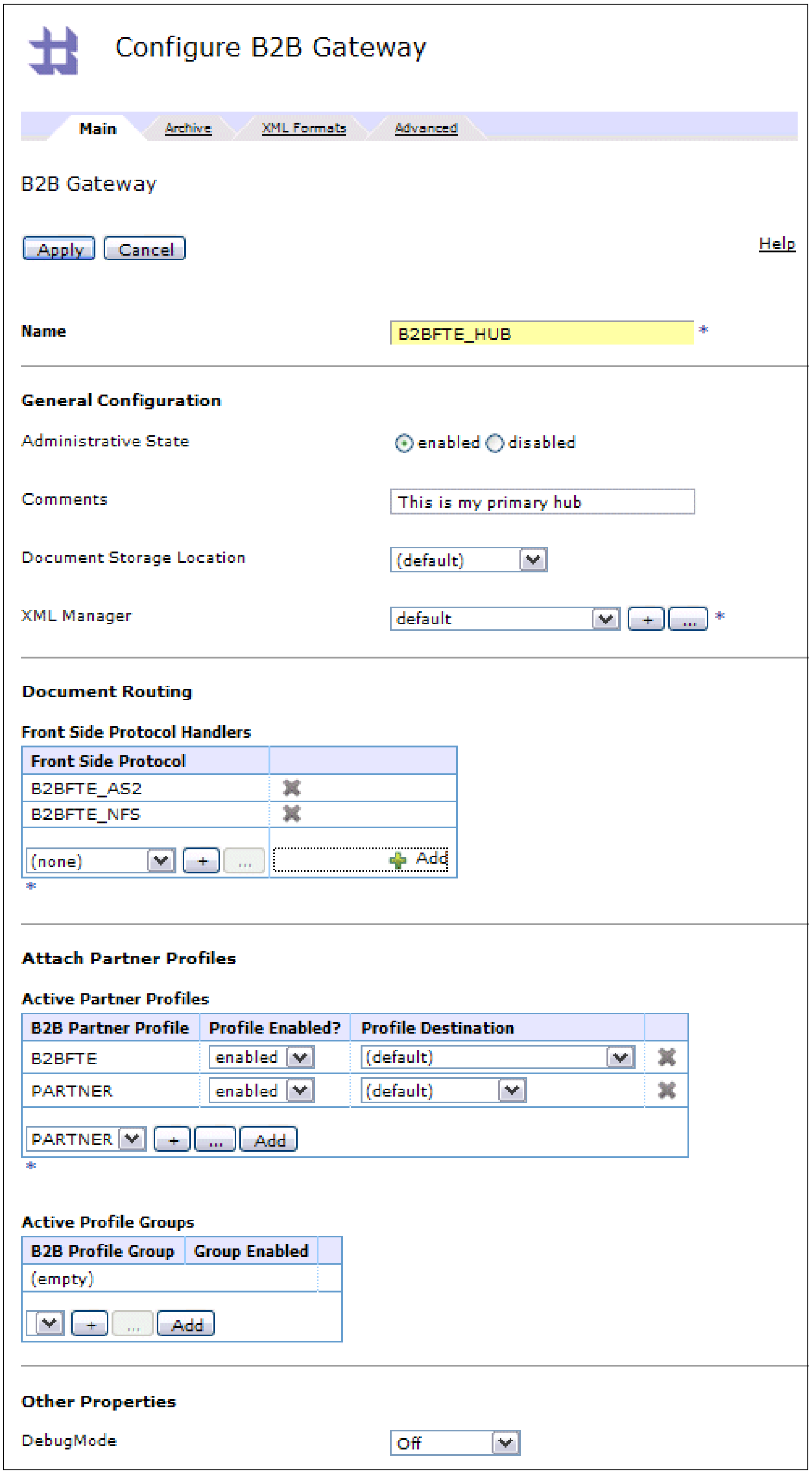
c. Accept the defaults for all of the other fields.

> **XML and Advanced tabs:** The XML tab is used to configure XPath locations of sender and receiver IDs in XML payloads. The Advanced tab is used to configure advanced B2B gateway properties, such as default URL fields for asynchronous MDNs, gateway priority, and the Document Routing Preprocessor to use for binary file routing. We are using EDI payloads for this scenario and have no need for XML XPaths and advanced functions, so we can skip the configuration of these two tabs.

5. Now that the B2B gateway is completely configured, save the service by clicking **Apply**.

6. Click **Save Config** to persist your changes.

Figure 7-47 illustrates how the fully configured B2BFTE_HUB B2B Gateway service should look. Fields that are populated with default values might not be displayed.



Configure B2B Gateway

B2B Gateway

Name: B2BFTE_HUB *

General Configuration

Administrative State: ● enabled ○ disabled

Comments: This is my primary hub

Document Storage Location: (default)

XML Manager: default *

Document Routing

Front Side Protocol Handlers

Front Side Protocol	
B2BFTE_AS2	✖
B2BFTE_NFS	✖
(none)	Add

Attach Partner Profiles

Active Partner Profiles

B2B Partner Profile	Profile Enabled?	Profile Destination	
B2BFTE	enabled	(default)	✖
PARTNER	enabled	(default)	✖
PARTNER	Add		

Active Profile Groups

B2B Profile Group	Group Enabled
(empty)	
	Add

Other Properties

DebugMode: Off

Figure 7-47 B2BFTE_HUB Gateway - Finished

7.5 Testing inbound and outbound flows

In this section we test our configuration of the XB60's B2B objects and WebSphere MQ File Transfer Edition using the flows described in 7.2.2, "Scenario flow inbound" on page 114, and 7.2.3, "Outbound data flow scenario" on page 115.

We simulate the partner's B2B gateway by using a separate domain on the same XB60 device instead of relying on a partner connection over the internet. The simulated partner's hub disposes of the payload and persists the metadata and all information needed to verify that the AS2 transaction is successful.

> **Download materials:** The exported PARTNER domain is available in the `Common_Files` directory provided in Appendix C, "Additional material" on page 317. You also must import the certificates for the PARTNER and B2BFTE profiles after importing the domain. The certificates can be found in the `B2BScenario_Files\certs` folder in Appendix C, "Additional material" on page 317.

7.5.1 Inbound AS2 flow testing

In this test case, we receive an EDI document from the partner's B2B gateway, PARTNERHUB, which wraps the EDI file in an AS2 header and sends it to the B2B gateway, B2BFTE_HUB. This B2B gateway receives the AS2 message, verifies security and partner information, and routes the payload to a multi-protocol gateway that integrates the XB60 to WebSphere MQ File Transfer Edition. After we have successfully written the file to the directory and sent the XML transfer command message, the XB60 sends a MDN back to the partner's B2B gateway informing it that the transfer was successful. To trigger the flow from the partner, we post the EDI file to a listener in the partner's gateway. This can be accomplished with any HTTP client utility, such as NetTool or CURL. Figure 7-48 and the text that follows describe the test for the inbound data flow in more detail.

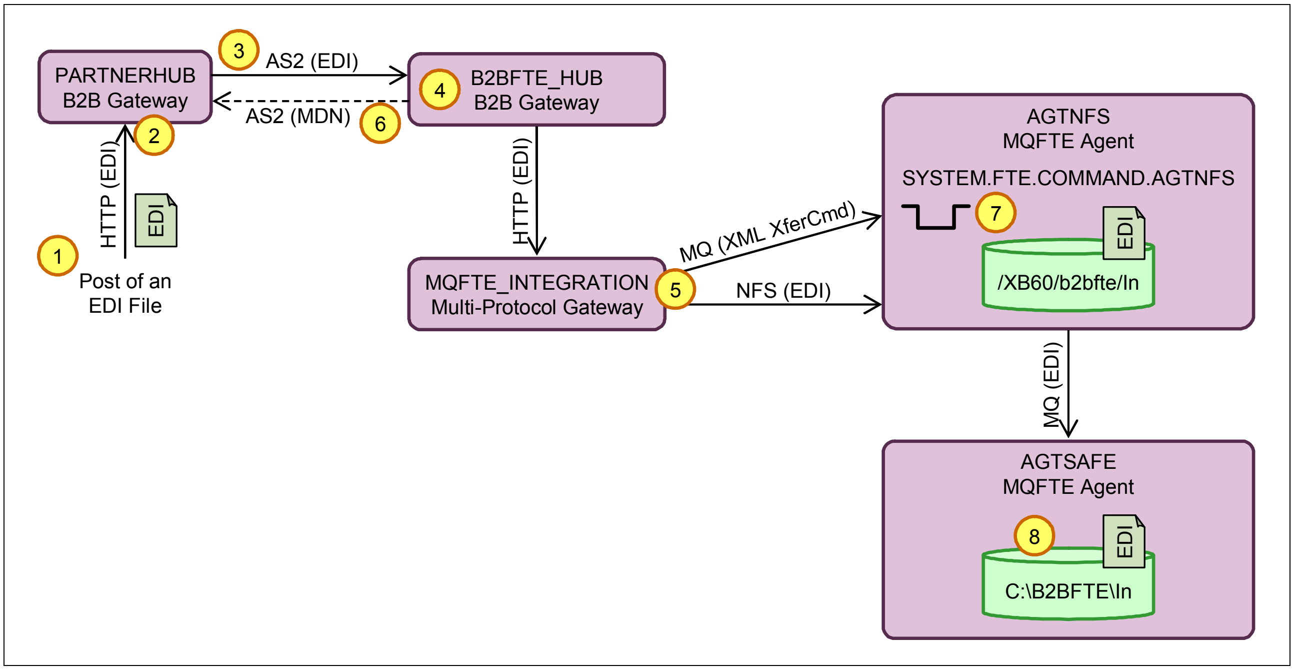


Figure 7-48 Inbound flow - Component view

The following steps describe the inbound test flow:

1. We use an HTTP client utility to post the sample EDI file to the simulated Partner’s HTTP listener.

   Additional material: The EDI file sent from the partner to the B2BFTE_HUB gateway for testing is included in the additional materials for this book. You can find it in `B2BScenario_Files\sample_files\partner_b2bfte.edi`. For more information, see Appendix C, “Additional material” on page 325. Uncompress this directory onto the desktop that is used to connect to the XB60 and post it to:

   `http://IP Address of the XB60:30003`

2. The PARTNERHUB B2B gateway parses the RAW EDI file, recognizes that it is EDI X12, and extracts the sender and receiver information from the file. Figure 7-49 shows where the sender and receiver information is located in the EDI file.

```
ISA*00*ASCENTIAL *01*92511930  *zz*partner        *zz*b2bfte
*940401*0942*U*00201*000000002*0*T*>
GS*PO*006250740*3122721850*940401*0942*1*X*002003
ST*850*1
BEG**BY*ab100**931028
NTE**This is a header message
SHH*DD*001*930701
N1*BT*Distributor Co
N3*2345 Waukegan Rd*E100
N4*Bannockburn*IL*60015*US
PO1*1*500*EA*45.26**IN*800-ABT1
NTE**Please paint this blue
PO1*1A*1000*EA*22.12**IN*900-ABT1
```

Figure 7-49 Sample Inbound EDI file - partner_b2bfte.edi

3. The PARTNERHUB B2B gateway wraps the file in an AS2 envelope and sends the AS2 packaged file to the B2BFTE_HUB B2B gateway.

4. The B2BFTE_HUB B2B gateway receives the AS2 message at the AS2 front side handler and extracts the sender and receiver information from the AS2 headers. Figure 7-50 shows where the sender and receiver information is located in the AS2 message.

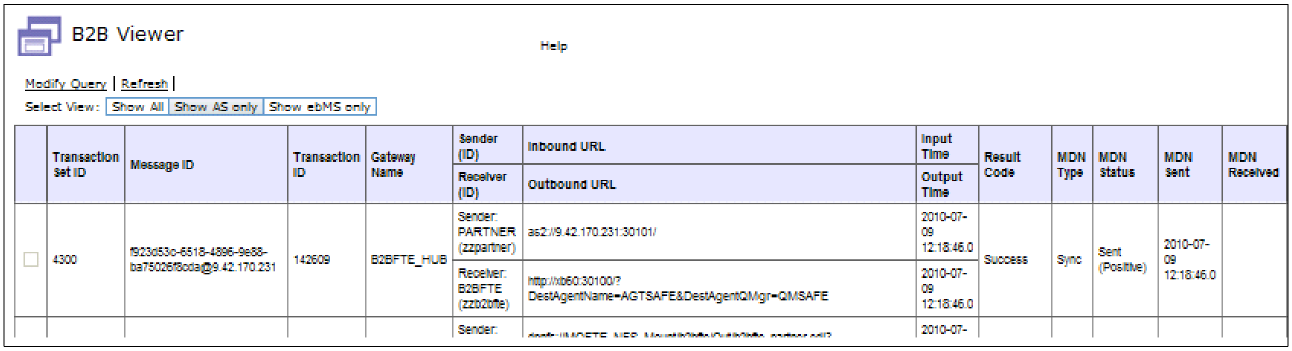
```
POST / HTTP/1.1
Host: xb60:30003
Cookie:
Via: 1.1 AQAAAGWFAHA=
X-CLIENT-IP: 213.98.90.21
Date: Tue, 27 Jan 2009 11:07:40 GMT
AS2-From: zzpartner
AS2-To: zzb2bfte
AS2-Version: 1.1
Message-ID: <ea7bf663-1539-4ca9-910e-e9b4c0025427@xb60>
Subject:  zzpartner To zzb2bfte
Disposition-Notification-To: ignored@example.com
Disposition-Notification-Options: signed-receipt-protocol=optional,
pkcs7-signature; signed-receipt-micalg=optional, sha1,md5
Recipient-Address: as2://xb60:30003/
```

Figure 7-50 Sample AS2 message envelope

5. The B2BFTE_HUB B2B gateway unpackages the AS2 Envelope, looks up the partner information, and verifies that the partners exist and are allowed to trade EDI documents. It looks at the destination that is configured for the receiving (internal) profile (B2BFTE), delivers the payload to the `/XB60/b2bfte/In` NFS directory, generates the XML transfer command XML message, and transfers it to the WebSphere MQ File Transfer Edition system command queue.
6. The B2BFTE_HUB B2B gateway sends an MDN to the partner to inform them that the AS2 message was successfully received.
7. WebSphere MQ File Transfer Edition reads the XML transfer command XML message and triggers a transfer between AGTNFS and AGTSAFE.
8. AGTSAFE writes the EDI file to the `C:\B2BFTE\In` directory.

Viewing the results

Now let us view the transaction in the XB60's B2B Transaction Viewer. Log on to the XB60, and click **B2B Transaction Viewer** from the menu. This displays the B2B Viewer panel (Figure 7-51).

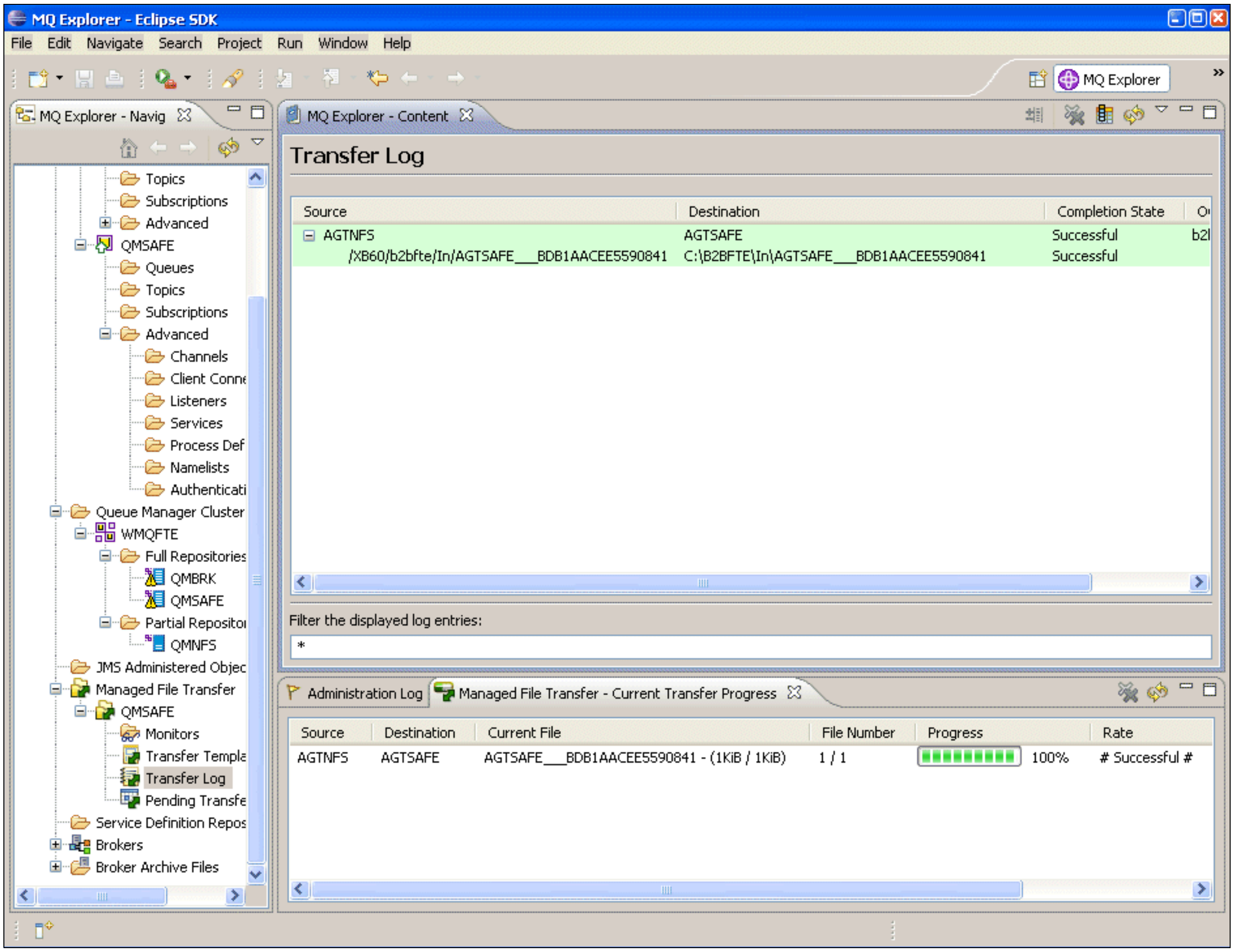


	Transaction Set ID	Message ID	Transaction ID	Gateway Name	Sender (ID)	Inbound URL	Input Time	Result Code	MDN Type	MDN Status	MDN Sent	MDN Received
					Receiver (ID)	Outbound URL	Output Time					
□	4300	f923d53c-6518-4896-9e88-ba75026f8cda@9.42.170.231	142609	B2BFTE_HUB	Sender: PARTNER (zzpartner)	as2://9.42.170.231:30101/	2010-07-09 12:18:46.0	Success	Sync	Sent (Positive)	2010-07-09 12:18:46.0	
					Receiver: B2BFTE (zzb2bfte)	http://xb60:30100/?DestAgentName=AGTSAFE&DestAgentQMgr=QMSAFE	2010-07-09 12:18:46.0					
					Sender:	[illegible]	2010-07-					

Figure 7-51 XB60 B2B Viewer - Inbound EDI to B2BFTE

In this example, you can see that #4300 is the inbound transaction from zzpartner destined for zzb2bfte and was processed by the B2B gateway named B2BFTE_HUB. You can see that the file represented by the message ID came into your AS2 front side handler and was sent to the multi-protocol gateway. You can also see that the gateway sent an MDN to the partner in the MDN Sent and MDN Status columns of the viewer. If you want to see the inbound, outbound, MDN, and content payload files, you can click the Transaction Set ID and choose the appropriate file.

Next we log on to the system running WebSphere MQ File Transfer Edition and launch the WebSphere MQ Explorer. On the Navigation bar we scroll down to Managed File Transfer and click **Transfer Log**. You will see the status of the transfer in the WebSphere MQ Explorer Content view and the progress view at the bottom of the panel. Figure 7-52 shows how the view should look.



Transfer Log

Source	Destination	Completion State	O
AGTNFS	AGTSAFE	Successful	b2l
/XB60/b2bfte/In/AGTSAFE___BDB1AACEE5590841	C:\B2BFTE\In\AGTSAFE___BDB1AACEE5590841	Successful	

Filter the displayed log entries:

*

Source	Destination	Current File	File Number	Progress	Rate
AGTNFS	AGTSAFE	AGTSAFE___BDB1AACEE5590841 - (1KiB / 1KiB)	1 / 1	100%	# Successful #

Figure 7-52 WebSphere MQ Explorer - inbound EDI from AGTNFS to AGTSAFE

In this example, you can see that the file was picked up by AGTNFS from the `/XB60/b2bfte/Out` directory and transferred to AGTSAFE, which wrote the file to `C:\B2BFTE\In` at the final destination. You can also see that the file was a success by looking at the completion state.

7.5.2 Outbound AS2 flow testing

In this test case, we use a feature in WebSphere MQ File Transfer Edition that allows us to monitor a directory using a polling cycle to trigger the outbound data flow. The monitor initiates the transfer of the EDI file. AGTSAFE reads the file from the source directory and transfers it to AGTNFS. AGTNFS writes the file to the target directory. The B2BFTE_HUB polls the target directory, finds the EDI file and picks it up, parses the file for sender and receiver information, verifies that the profiles are allowed to trade EDI files, wraps the EDI file in AS2 based on the destination configuration in the external partner profile, and sends the AS2 message to the external partner. Upon receipt of the file, the external partner sends an MDN back to the B2BFTE_HUB gateway indicating that the transfer was successful. Figure 7-53 and the text that follows describe the outbound data flow in more detail.

Simulating the partner: The simulated partner configuration in the PARTNER domain of the XB60 is designed to consume the EDI payload and throw it away. In a real-world scenario your external partner would route the EDI payload to the appropriate systems.

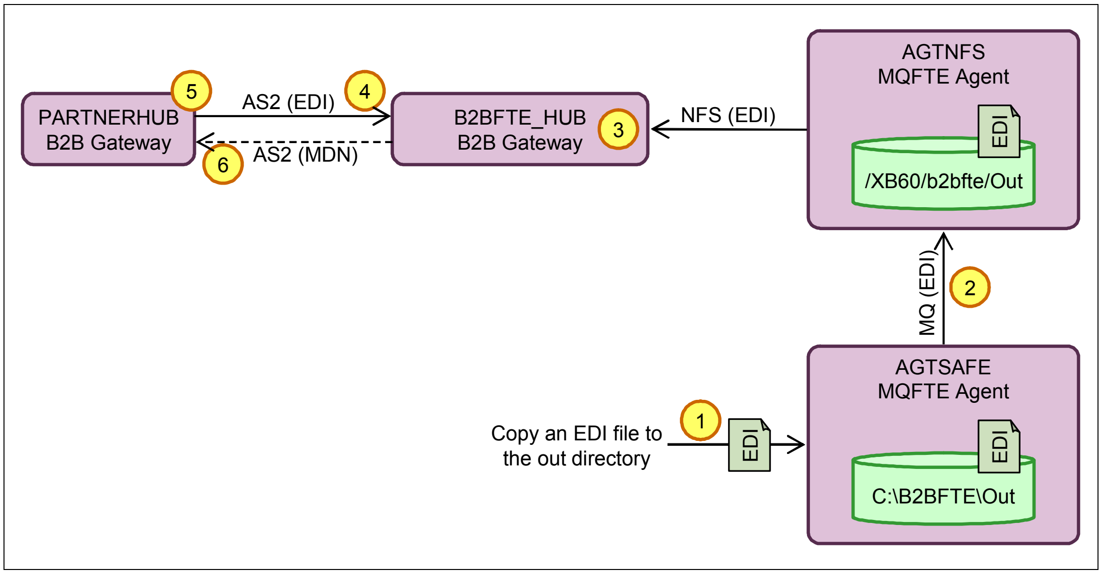


Figure 7-53 Outbound flow - Component view

The following steps describe the outbound test flow:

1. The sample EDI file is copied to the directory that is being monitored by WebSphere MQ File Transfer Edition. The monitor was configured in 7.3.6, “Creating a monitor to poll the outbound directory” on page 121.

   Additional material: The EDI file transferred from the B2BFTE_HUB gateway to the partner can be found in Appendix C, “Additional material” on page 317. The `b2bfte_partner.edi` file is located in the `B2BScenario_Files\sample_files` directory. Place this directory on the desktop being used to connect to the XB60.

2. The monitor triggers the outbound flow based on the presence of the EDI file in the directory that it is monitoring. AGTSAFE picks up the file from `C:\B2BFTE\Out` and transfers the file to AGTNFS, which writes the file to the `/XB60/b2bfte/Out` directory.

3. The B2BFTE_HUB polls the NFS directory, picks up and parses the RAW EDI file, recognizes that it is EDI X12, and extracts the sender and receiver information from the file. Figure 7-54 shows where the sender and receiver information is located in the EDI file.

```
ISA*00*ASCENTIAL *01*92511930  *zz*b2bfte        *zz*partner
*940401*0942*U*00201*000000002*0*T*>
GS*PO*006250740*3122721850*940401*0942*1*X*002003
ST*850*1
BEG**BY*ab100**931028
NTE**This is a header message
SHH*DD*001*930701
N1*BT*Distributor Co
N3*2345 Waukegan Rd*E100
N4*Bannockburn*IL*60015*US
PO1*1*500*EA*45.26**IN*800-ABT1
NTE**Please paint this blue
PO1*1A*1000*EA*22.12**IN*900-ABT1
```

Figure 7-54 Sample outbound EDI file - b2bfte_partner.edi

4. The B2BFTE_HUB gateway wraps the file in an AS2 envelope based on the external partner profile destination and sends the AS2 packaged file to the PARTNERHUB B2B gateway. Figure 7-55 shows where the sender and receiver information is located in the AS2 message.

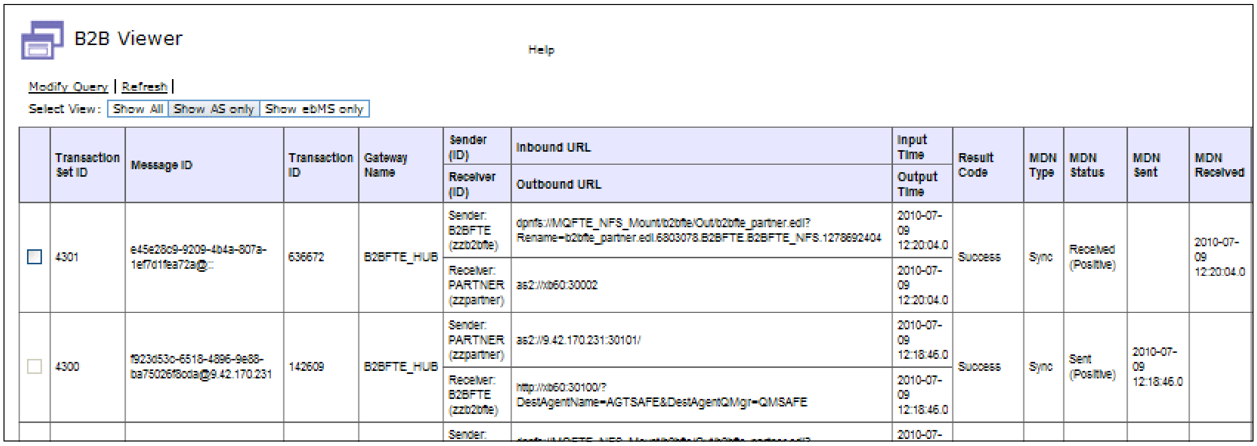
```
POST / HTTP/1.1
Host: xb60:30002
Cookie:
Via: 1.1 AQAAAGWFAHA=
X-CLIENT-IP: 213.98.90.21
Date: Tue, 27 Jan 2009 11:08:40 GMT
AS2-From: zzb2bfte
AS2-To: zzpartner
AS2-Version: 1.1
Message-ID: <fx3bf663-25t9-4ca9-910e-e9b4c023456@xb60>
Subject:  zzpartner To zzb2bfte
Disposition-Notification-To: ignored@example.com
Disposition-Notification-Options: signed-receipt-protocol=optional,
pkcs7-signature; signed-receipt-micalg=optional, sha1,md5
Recipient-Address: as2://xb60:30002/
```

Figure 7-55 Sample outbound AS2 message envelope

5. The PARTNERHUB B2B gateway unpackages the AS2 envelope, looks up the partner information, and verifies that the partner profiles exist and are allowed to trade EDI documents. It looks at the destination that is configured for the receiving profile and delivers the payload. In this scenario we dispose of the payload.

6. The PARTNERHUB sends an MDN to your B2BFTE_HUB gateway to inform you that the AS2 message was successfully received.

Now let us view the transaction in the XB60's B2B transaction viewer. Log on to the XB60 and click **B2B Transaction Viewer** from the menu. This displays the B2B Viewer panel (Figure 7-56).



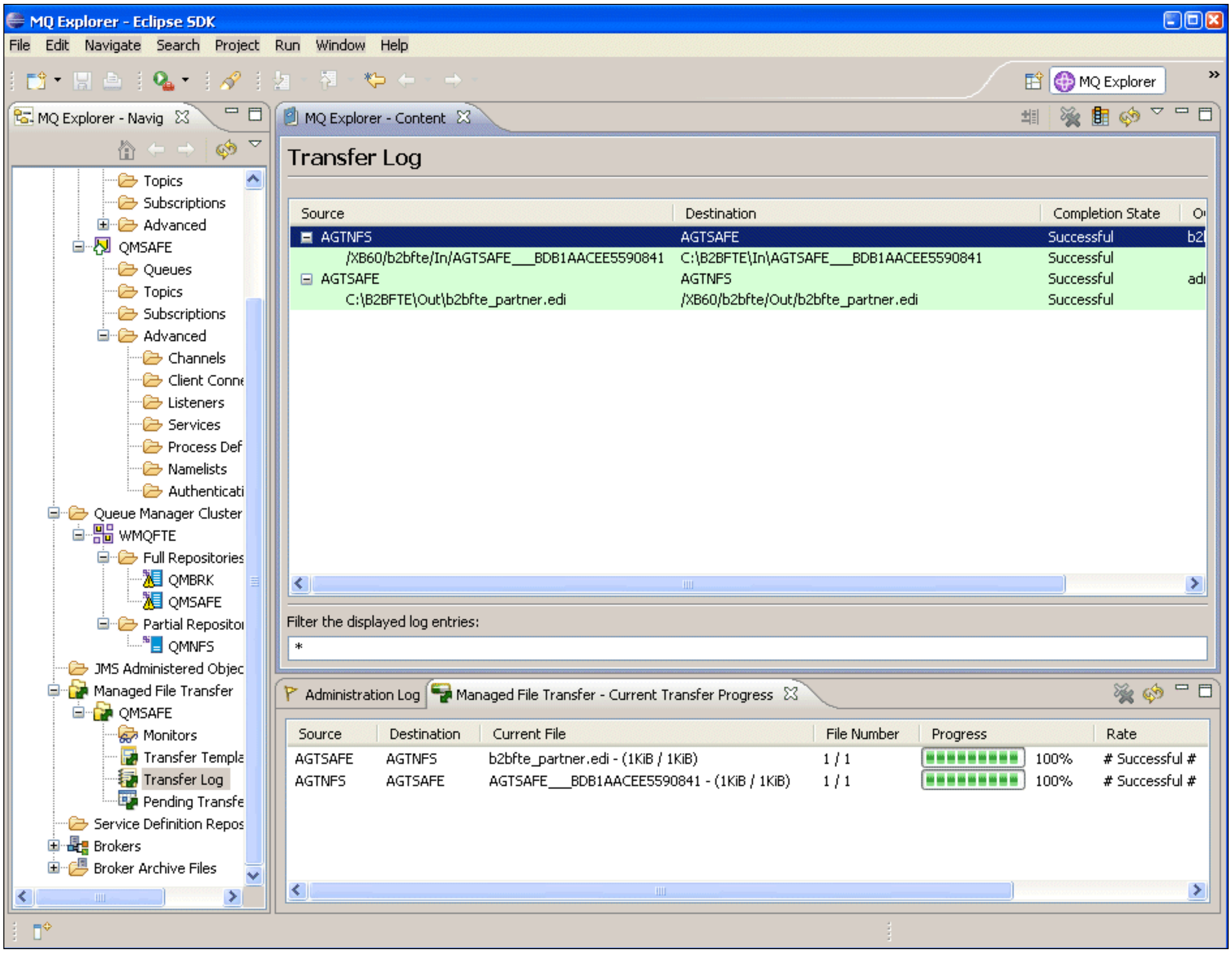
B2B Viewer

	Transaction Set ID	Message ID	Transaction ID	Gateway Name	Sender (ID)	Inbound URL	Input Time	Result Code	MDN Type	MDN Status	MDN Sent	MDN Received
					Receiver (ID)	Outbound URL	Output Time					
□	4301	e45e28c9-9209-4b4a-807a-1ef7d1fea72a@::	636672	B2BFTE_HUB	Sender: B2BFTE (zzb2bfte)	dpnfs://MQFTE_NFS_Mount/b2bfte/Out/b2bfte_partner.edi?Rename=b2bfte_partner.edi.6803078.B2BFTE.B2BFTE_NFS.1278692404	2010-07-09 12:20:04.0	Success	Sync	Received (Positive)		2010-07-09 12:20:04.0
					Receiver: PARTNER (zzpartner)	as2://xb60:30002	2010-07-09 12:20:04.0					
□	4300	f923d53c-6518-4896-9e88-ba75026f8cda@9.42.170.231	142609	B2BFTE_HUB	Sender: PARTNER (zzpartner)	as2://9.42.170.231:30101/	2010-07-09 12:18:46.0	Success	Sync	Sent (Positive)	2010-07-09 12:18:46.0	
					Receiver: B2BFTE (zzb2bfte)	http://xb60:30100/?DestAgentName=AGTSAFE&DestAgentQMgr=QMSAFE	2010-07-09 12:18:46.0					
					Sender:		2010-07-					

Figure 7-56 B2B transaction viewer

In this example, you can see that #4301 was the outbound transaction sent from zzb2bfte to zzpartner, and the file was processed by the B2B gateway named B2BFTE_HUB. You can see that the file came from the NFS file share and was sent to the external partner using an AS2 protocol and destination.You can also see that you received an MDN from the partner in the MDN Received and MDN Status column. If you want to see the inbound, outbound, MDN, and content files, you can click the Transaction Set ID and choose the appropriate file.

Next log on to the system running WebSphere MQ File Transfer Edition and launch WebSphere MQ Explorer. On the Navigation bar scroll down to Managed File Transfer and click **Transfer Log**. You will see the status of the transfer in the WebSphere MQ Explorer Content view and the progress view at the bottom of the panel. Figure 7-57 shows how the view should look.



Transfer Log

Source	Destination	Completion State	O
⊟ AGTNFS	AGTSAFE	Successful	b2
/XB60/b2bfte/In/AGTSAFE___BDB1AACEE5590841	C:\B2BFTE\In\AGTSAFE___BDB1AACEE5590841	Successful	
⊟ AGTSAFE	AGTNFS	Successful	ad
C:\B2BFTE\Out\b2bfte_partner.edi	/XB60/b2bfte/Out/b2bfte_partner.edi	Successful	

Filter the displayed log entries:

*

Source	Destination	Current File	File Number	Progress	Rate
AGTSAFE	AGTNFS	b2bfte_partner.edi - (1KiB / 1KiB)	1 / 1	100%	# Successful #
AGTNFS	AGTSAFE	AGTSAFE___BDB1AACEE5590841 - (1KiB / 1KiB)	1 / 1	100%	# Successful #

Figure 7-57 WebSphere MQ Explorer - Outbound EDI from AGTSAFE to AGTNFS

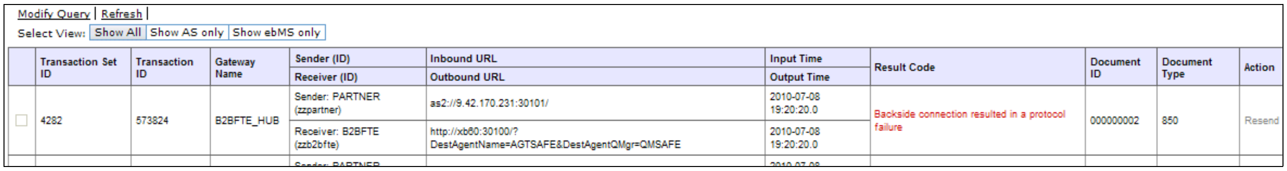
In this example, you can see that the file was picked up by AGTSAFE from the `C:\B2BFTE\Out` directory and transferred to AGTNFS. AGTNFS wrote the file to the `/XB60/b2bfte/Out` directory, which is being polled by the XB60 B2BFTE_HUB gateway. You can also see that the file was a success by looking at the completion state.

7.6 Troubleshooting tips

This section describes the common configuration mistakes, and that can cause the configuration in this scenario to fail.

7.6.1 Protocol failure error in the viewer

The protocol failure error (Figure 7-58) is typically caused when the XB60 cannot reach the back side of the flow. For this scenario the back side is the destination of the internal partner profile (B2BFTE).



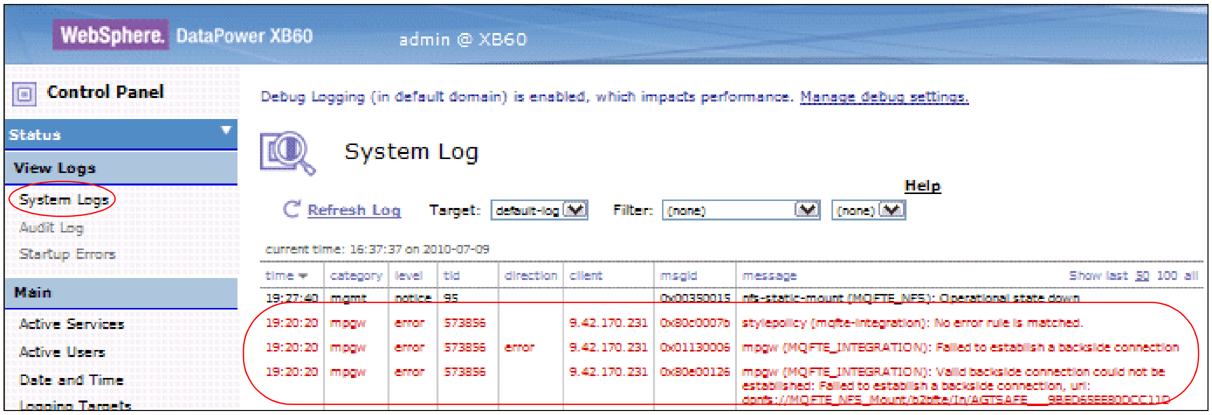
Modify Query | Refresh |

Select View: Show All | Show AS only | Show ebMS only

	Transaction Set ID	Transaction ID	Gateway Name	Sender (ID) Receiver (ID)	Inbound URL Outbound URL	Input Time Output Time	Result Code	Document ID	Document Type	Action
□	4282	573824	B2BFTE_HUB	Sender: PARTNER (zzpartner) Receiver: B2BFTE (zzb2bfte)	as2://9.42.170.231:30101/ http://xb60:30100/?DestAgentName=AGTSAFE&DestAgentQMgr=QMSAFE	2010-07-08 19:20:20.0 2010-07-08 19:20:20.0	Backside connection resulted in a protocol failure	000000002	850	Resend
				Sender: PARTNER		2010-07-08				

Figure 7-58 B2B Viewer - Protocol failure

On the inbound flow we use the multi-protocol gateway service as a post process to integrate to WebSphere MQ File Transfer Edition. Because the viewer is meant to be an at-a-glance view of the state of the transaction, we need to get more details about the error. To do this we look into the XB60's System Log. Do this by expanding the **Status** drop-down menu in the XB60 navigation menu and clicking **System Logs**. Figure 7-59 shows the error in the XB60 System Log. Notice that the log is more verbose.



Debug Logging (in default domain) is enabled, which impacts performance. Manage debug settings.

Target: default-log Filter: (none) (none)

current time: 16:37:37 on 2010-07-09

time	category	level	tid	direction	client	msgid	message Show last 50 100 all
19:27:40	mgmt	notice	95			0x00350015	nfs-static-mount (MQFTE_NFS): Operational state down
19:20:20	mpgw	error	573856		9.42.170.231	0x80c0007b	stylepolicy (mqfte-integration): No error rule is matched.
19:20:20	mpgw	error	573856	error	9.42.170.231	0x01130006	mpgw (MQFTE_INTEGRATION): Failed to establish a backside connection
19:20:20	mpgw	error	573856		9.42.170.231	0x80e00126	mpgw (MQFTE_INTEGRATION): Valid backside connection could not be established: Failed to establish a backside connection, url: dpnfs://MQFTE_NFS_Mount/b2bfte/In/AGTSAFE___9BED65EE80DCC11D

Figure 7-59 XB60 System Log - Protocol failure

You can see from the log that the multi-protocol gateway received the file, but it did not deliver it to the NFS mount point. Use the following steps to troubleshoot this error.

1. Look at the NFS mount point that you configured in 7.4.3, "Configuring a static NFS mount object" on page 124:

   a. If it is up, then proceed to step 2.

   b. If it is down, check to be sure that the server running NFS is accessible and that all of the directories have the appropriate permissions. Also, check to be sure that the Remote NFS Export field is filled in with the correct information.

2. Look at the multi-protocol gateway processing policy that you configured in 7.4.5, "Configuring a multi-protocol gateway for WebSphere MQ File Transfer Edition integration" on page 128. Open the policy and look at the transform rule on the Advanced tab.

   Be sure that the DestinationURL parameter has the proper value in it. This value must point to the DataPower mount point and the exact directory under that mount point. Linux is case sensitive, so be sure that the case matches the exact directory structure created on the Linux system. The most common area for error is forgetting to add the trailing front slash at the end of the URL.

Tip: In this scenario, the correct value for the DestinationURL parameter is:

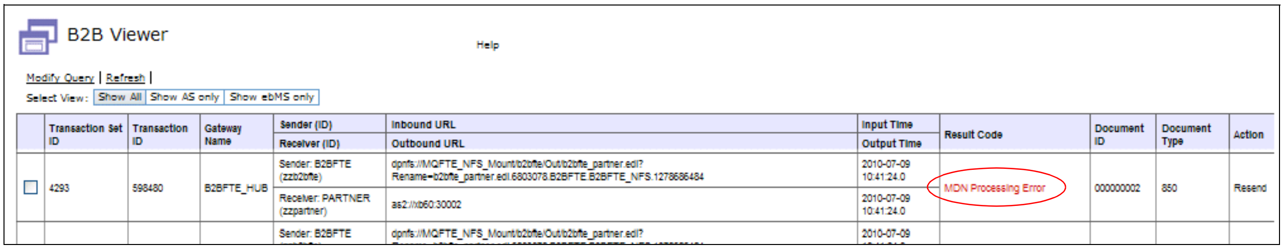
```
dpnfs://MQFTE_NFS_Mount/b2bfte/In/
```

After resolving any issues related to the steps above you now should be able to successfully receive an inbound file.

Tip: This same error is seen in the B2B viewer if the a queue manager is down. However, the log tells you that the queue manager cannot be reached. The troubleshooting steps are the same for the queue manager, except that you look at the MQ Queue Manager object in the XB60, the system running WebSphere MQ File Transfer Edition, and in the Multi-Protocol processing policy you look at the MQFTENotificationTarget parameter.

7.6.2 MDN and processing error

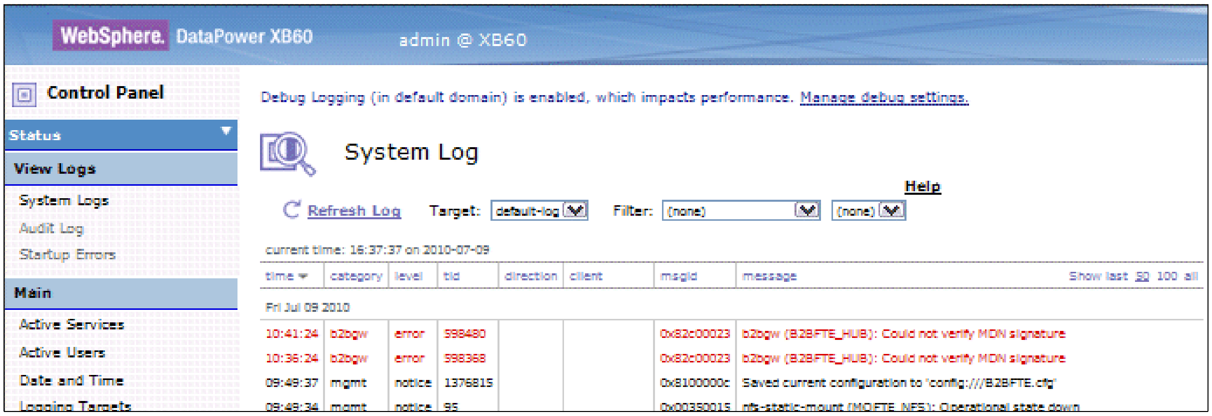
The MDN processing error (Figure 7-59 on page 171) typically is caused when the XB60 has an issue with the MDN that was received from the external partner. It can be due to a malformed MDN or problems with the signature used for the MDN.



	Transaction Set ID	Transaction ID	Gateway Name	Sender (ID) Receiver (ID)	Inbound URL Outbound URL	Input Time Output Time	Result Code	Document ID	Document Type	Action
□	4293	598480	B2BFTE_HUB	Sender: B2BFTE (zzb2bfte) Receiver: PARTNER (zzpartner)	dpnfs://MQFTE_NFS_Mount/b2bfte/Out/b2bfte_partner.edi?Rename=b2bfte_partner.edi.6803078.B2BFTE.B2BFTE_NFS.1278686484 as2://xb60:30002	2010-07-09 10:41:24.0 2010-07-09 10:41:24.0	MDN Processing Error	000000002	850	Resend
				Sender: B2BFTE	dpnfs://MQFTE_NFS_Mount/b2bfte/Out/b2bfte_partner.edi?	2010-07-09				

Figure 7-60 MDN Processing Error - B2B Viewer

Because we are requesting signed MDNs, getting this error in the outbound flow likely means that something is wrong with our signature. Because the viewer is meant to be an at-a-glance view of the state of the transaction, we need to get more details about the error. To do this we look into the XB60's System Log. Do this by expanding the **Status** drop-down menu in the XB60 navigation menu and clicking **System Logs**. Figure 7-61 shows you how the error looks in the XB60 System Log.



Debug Logging (in default domain) is enabled, which impacts performance. Manage debug settings.

current time: 16:37:37 on 2010-07-09

time	category	level	tid	direction	client	msgid	message
Fri Jul 09 2010							
10:41:24	b2bgw	error	598480			0x82c00023	b2bgw (B2BFTE_HUB): Could not verify MDN signature
10:36:24	b2bgw	error	598368			0x82c00023	b2bgw (B2BFTE_HUB): Could not verify MDN signature
09:49:37	mgmt	notice	1376815			0x8100000c	Saved current configuration to 'config:///B2BFTE.cfg'
09:49:34	mgmt	notice	95			0x00350015	nfs-static-mount (MQFTE_NFS): Operational state down

Figure 7-61 MDN Processing Error - XB60 System Log

You can see from the system log that we were unable to verify the signature that was applied to the MDN that we received. Because the partner is responsible for signing outbound documents with its private key, this error likely means that we are not using the correct certificate in the PARTNER profile in our B2BFTE domain. Check the profile's AS Security tab in the inbound security section and be sure that you are using the partner's public certificate to verify signatures. The certificate that we need to use is partner-sscert.pem.

After you resolve the certificate error, you should be able to successfully send outbound files to PARTNER and process the signed MDN.

8

Integrating partner transfers with internal ESB

In this chapter we outline another common scenario for transferring files between organizations. This scenario uses a gateway, but does not use the typical business-to-business (B2B) protocols. Instead we show how to implement a gateway with a common application layer protocol. This scenario uses partner profile management at the edge of the network and a file transfer backbone inside the protected network to move files securely between external partners and back-end systems.

This scenario integrates the IBM WebSphere DataPower B2B Appliance XB60 with WebSphere Message Broker over the WebSphere MQ File Transfer Edition backbone network (referred to as the MQ FTE backbone) to implement the solution.

Additional material: Additional material is provided in support of the activities in this chapter. If you have access to the components needed to build this solution and want to try this on your own, the additional material can be downloaded to assist you with your configuration and with testing the setup. For more information about downloading the additional material and its contents, see Appendix C, “Additional material” on page 317. The files for this chapter can be found in the `Common_Files` and the `BrokerScenario_Files` directories.

8.1 Scenario overview

This scenario shows how we can use the XB60's partner profile management to validate partners connecting with protocols other than business-to-business (B2B) messaging protocols. We also show how to integrate the XB60 with the MQ FTE backbone network to connect to an organization's enterprise service bus (ESB) for routing requests. Connecting to an ESB not only provides the ability to route data to the appropriate back-end application, but also allows data to be transformed appropriately for that application. Protocols can be converted to one that the application understands, whether that is an MQ message, file, or another WebSphere MQ File Transfer Edition transfer.

With this pattern, an external partner can send a request using any protocol that the XB60 supports. In this scenario, we use Secure File Transfer Protocol (SFTP) because it is a prevalent non-business-to-business protocol. The XB60 can act as an SFTP server with a front side handler in the B2B gateway.

We want the SFTP clients to connect through the B2B gateway, because it is this gateway that provides security and governance at the edge of the network with partner profile management and authentication, authorization and auditing (AAA) security. In this instance, the B2B gateway performs profile management by verifying that the connecting client is a valid partner in the gateway. SFTP is an application layer protocol that provides a sufficient level of transport and connection security. However, it does not provide data security and non-repudiation like a business-to-business messaging protocol does. After validating the partner, the B2B gateway forwards the payload to the multi-protocol gateway being used for integration with WebSphere MQ File Transfer Edition.

The multi-protocol gateway invokes the file transfer from a source agent to a target agent in WebSphere Message Broker. A message flow in the broker provides some file mediation and processing before passing the file to the backend application.

8.1.1 Appropriate use

This scenario demonstrates SFTP communications between an external partner and the B2B gateway representing the internal partner. However, note that this scenario can be varied to use any of the application layer messaging protocols supported on the XB60. At the time of writing, XB60 v3.8.1.2 firmware supports the following application layer protocols:

- FTPs
- Hypertext Transfer Protocols (HTTPs)
- SFTP
- Post Office Protocol 3 (POP3)
- MQ
- Java Message Service (JMS)
- Network File System (NFS)
- IMS Connect
- TIBCO EMS

To vary the scenario to utilize one or more of these protocols, simply configure front-side protocol handlers for each protocol and associate the handlers with the B2B gateway. You might also need to configure external partner destinations to utilize the same protocols.

There are many ways to vary the WebSphere MQ File Transfer Edition deployment to meet specific needs by using more sophisticated topologies. These topologies are described in detail in *Getting Started with WebSphere MQ File Transfer Edition V7*, SG24-7760.

8.1.2 Business value

There is significant business value in integrating the XB60 into the MQ FTE backbone to enable reliable and auditable internal file transfers and secure external file transfers between organizations by providing partner profile management and security at the edge of the network. The addition of WebSphere Message Broker to the MQ FTE backbone provides a mediation point for file processing as the file travels to the back-end applications.

Multi-enterprise integration with business partners is an important requirement for many companies, yet companies sometimes experience barriers to implementing a full business-to-business platform even if they have a solution that supports robust business-to-business functions. Many companies prefer to deploy their partner connections in a phased approach, in which they start by connecting over standard application layer protocols, then migrate the partners to a more sophisticated business-to-business type of connection at some point in the future. This scenario illustrates how the XB60, WebSphere MQ File Transfer Edition, and WebSphere Message Broker can provide the key elements in file transfer solutions that provide a framework for a phased approach for connecting to partners. It provides an integration backbone that does not have to change when the company decides to move partners from a application layer transfer protocol to a business-to-business protocol.

Multi-enterprise integration means moving files from an external partner's application to an internal back-end application with minimal to no human intervention, yet there is often a requirement for users to be able to see the status and history of the transfers. Using the XB60 with profile management and an MQ FTE backbone network gives you the ability to track all file transfers between external partners and downstream systems.

Administrators want to control which partners they trade with and which protocols they use to consume files from each partner. Business-to-business protocols and standards might not be appropriate for specific reasons from specific partners, but the requirement for exchanging files in a reliable and secure manner still exists. The XB60 provides the capability to exchange files using a variety of protocols, while still providing the partner profile management that you expect in a business-to-business deployment.

The following list describes the combined benefits that you can expect from this type of deployment scenario:

- Security
  - The XB60 provides exceptional data security and certificate management with robust authentication, authorization, and auditing capabilities. It has the built-in capability to integrate with external repositories.
  - WebSphere Message Broker and internal applications do not need to authenticate, because the XB60 is a trusted connection.
  - Partner profile management is provided by XB60, relieving WebSphere Message Broker and internal applications from having to perform partner verification.
  - The XB60 terminates external partner connections in the DMZ. Clients do not have access to the protected network where WebSphere Message Broker and internal applications are running.
  - All files persisted to the XB60's internal drive are Advanced Encryption Standard (AES) encrypted to ensure that no sensitive data is at rest in the DMZ that is not appropriately protected.

  - Access to files is controlled by file system permissions utilizing role-based management and access control configuration in the XB60.
  - File transfers between external partners can be protected using SSL encryption and authentication in the XB60. Connections into the protected network can also be configured with SSL using Message Broker.
- Administration, operations, and logging
  - Having the ability to trace the file transfers end-to-end with the XB60 and WebSphere MQ File Transfer Edition reduces the resources required to troubleshoot file transfer failures and retries.
  - WebSphere MQ File Transfer Edition allows you to set up file transfers to occur at specified times or dates, or to be repeated at specified intervals. File transfers can also be triggered by a range of system events, such as new files or updated files.
  - The XB60 provides the ability for external partners to view the state of their own transactions, and they have the ability to manually re-send transactions to themselves.
  - The XB60 reduces the need for in-house skills that are typically needed to deploy and manage business-to-business enabled file transfer solutions.
  - The XB60 provides robust logging and support for saving logs to a large variety of log targets utilizing a broad range of log formats.
  - WebSphere MQ File Transfer Edition provides full logging of transfers at both the source and destination systems for internal transfers.

8.2 Scenario details

This scenario shows multi-enterprise integration with an internal back-end application. We highlight the fact that a file transfer over SFTP can be sent from external partners to the XB60 acting as an SFTP server. This use case does not use business-to-business messaging protocols. However, we are still able to use the partner profile management in the XB60 B2B gateway to validate trading partners. The XB60 can integrate with WebSphere Message Broker and back-end applications via the MQ FTE backbone network.

In this scenario, we use the example of an external partner that wants to order stock from the company represented in the Figure 8-2 on page 182 as the internal partner. The file being sent over SFTP represents a stock replenishment request. The scenario shows how the request is routed through the B2B gateway to the NFS file server, then transferred to the broker system, so that the message flow can route the request to the order inventory system, and also route it to an archival system for storage in a history database. This scenario is not concerned with the application processing of the request. Our task is to examine how the request gets to that application.

We expect the order inventory system to return a response to the customer with a stock availability quantity, the availability of any promotions or pricing discounts, and the order status. Again, we are not concerned with the back-end application processing, but how that application sends the response to the external partner. This scenario assumes that the application’s response is put on an MQ queue that is defined in a WebSphere Message Broker MQInput node.

We examine how the response is transferred through the internal systems, from the application to WebSphere Message Broker for mediation before WebSphere MQ File Transfer Edition transfers it to the XB60 B2B gateway and finally out of the internal network to the external partner over SFTP.

8.2.1 Solution components

This section describes the components associated with each product in the solution. Certain components require specific configuration for the solution to be accomplished. We discuss the configuration steps required where necessary. Figure 8-1 shows the components.

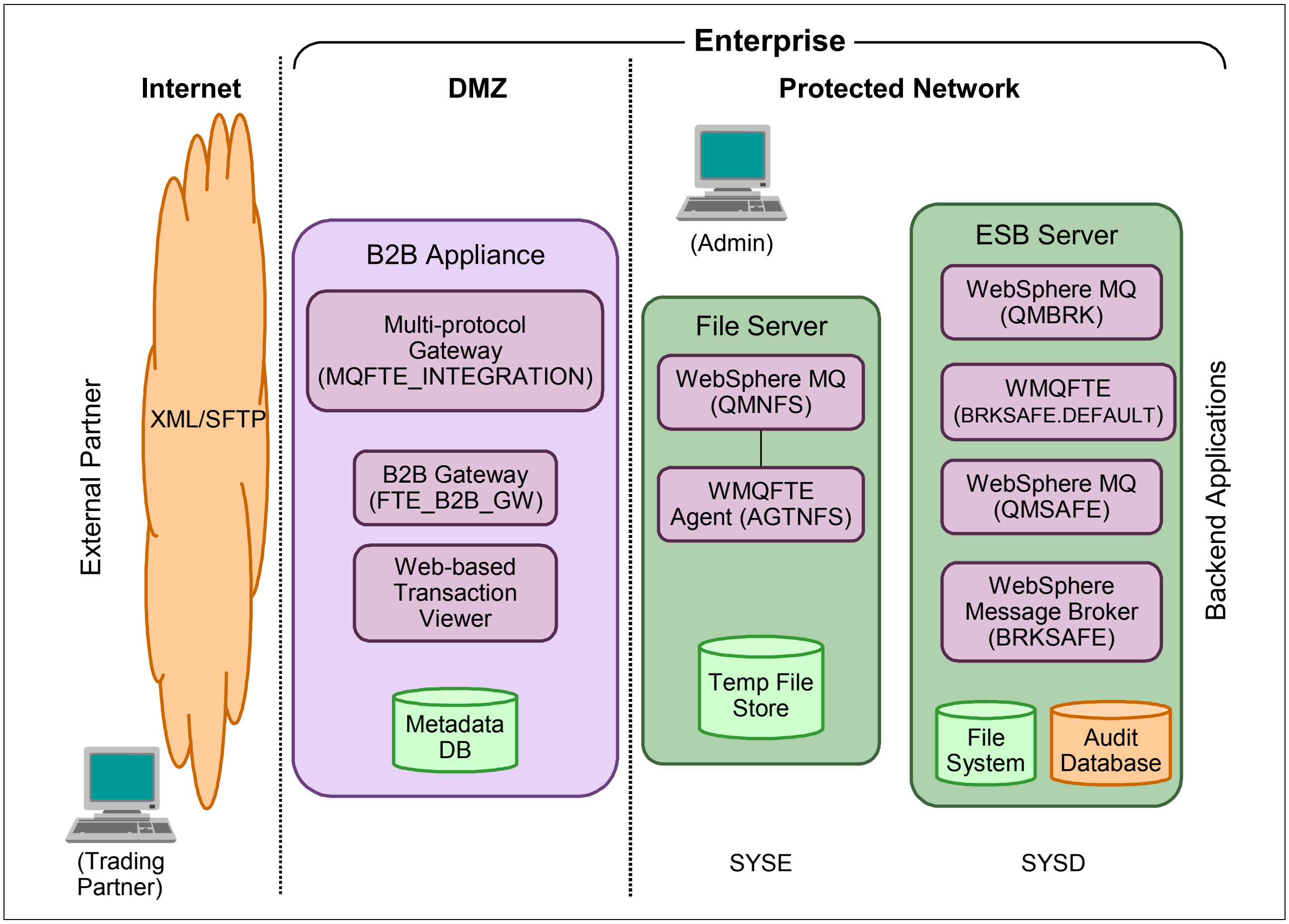


Figure 8-1 Solution components

XB60 configuration objects

This section describes each configuration object used in the XB60 to support this scenario. The XB60 provides the ability to configure the appliance utilizing the following options:

- Web-based graphical user interface (GUI)
- A command-line interface (CLI)
- An XML SOAP management interface

For the purpose of this scenario we use the web-based GUI.

External partner profile

External partner profiles represent the companies that do business with you. If your external partner is using profile management, it should complete a configuration process on its B2B gateway systems to configure an external representation of your connection profile. In most cases, the partner simply uses an SFTP server, on which you put files that are meant for it, and it puts files on the XB60 that are destined for systems in your protected network. Partner information in the payload is used to verify a valid inbound partner connection when using profile management. External partners might or might not have the ability to do the same level of partner verification. Our outbound flow does not concern itself with how the external partner processes the data that we send it. It is up to the external partner as to how to securely move the file into its protected network.

In this scenario the external partner's company is emulated by a separate application domain on the XB60 named PARTNER, with a B2B Gateway service named PARTNERHUB. The external partner profile in this scenario is represented in the FTE_B2B_GW gateway by a profile object named Partner.

Internal partner profile

The internal partner profile in the XB60 typically is associated with your company or a department inside your company. Your company typically is responsible for the purchase and construction of the B2B gateway product, including definition of the electronic business processes transacted between your company and your external partners. The internal partner's company is represented by a B2B Gateway service named FTE_B2B_GW. The internal partner profile in this scenario is represented in the FTE_B2B_GW gateway by a profile object named FTEBROKER.

B2B Gateway service

The B2B Gateway service is a configuration object that is responsible for processing and routing business-to-business data. All data that passes through the B2B Gateway service is tied to profile management, meaning that partner information is either extracted from the payload or it is set using XSLT. This service automatically extracts partner IDs from XML utilizing XPaths configured in the B2B Gateway service. Binary files cannot be parsed for partner IDs and must use the document routing preprocessor in the B2B Gateway service to set the IDs. Additionally, all data that passes through this service is persisted to a hard drive for non-repudiation, and all metadata associated with the transaction is persisted in a data store and is visible to the user in the B2B transaction viewer. The B2B Gateway service in this scenario is named FTE_B2B_GW.

Multi-protocol gateway service

The multi-protocol gateway service is a configuration object that is responsible for processing and routing any data type. It is essentially a secure router/firewall type of service. Unlike the B2B Gateway service, this service does not persist any data and is not tied to profile management or to the B2B viewer. The multi-protocol gateway service in this scenario is named MQFTE_INTEGRATION and is used as a post-process attached to the B2B Gateway service used to integrate with WebSphere MQ File Transfer Edition.

B2B transaction viewer

The B2B transaction viewer is used to view all transactions that pass through a B2B gateway service. The viewer can be used to manually re-send transactions from the XB60 to external partners. Additionally, you can allow an external partners access to view the state of its transactions by using the appliance’s access control and role-based management capabilities.

WebSphere MQ and WebSphere MQ File Transfer Edition components

This section describes each WebSphere MQ and WebSphere MQ File Transfer Edition component used to support this scenario. This scenario uses three queue managers and three agents.

WebSphere MQ queue managers

Queue manager QMNFS hosts queues used by the WebSphere MQ File Transfer Edition AGTNFS agent on the internal file server.

Queue manager QMSAFE is the coordination queue manager for the internal MQ FTE backbone network. The coordination queue manager publishes status messages received from the agents. QMSAFE also hosts the queues for the database logger (DBLOGGER) application, which subscribes to publications from the QMSAFE and stores them in a DB2 database.

Queue manager QMBRK also runs on SYSD, the WebSphere Message Broker server. QMBRK hosts queues used by the broker and the WebSphere MQ File Transfer Edition agent that runs in the broker execution group.

WebSphere MQ File Transfer Edition agents

AGTNFS is a WebSphere MQ File Transfer Edition agent on SYSE, the internal file server. It connects to the local queue manager QMNFS via bindings mode. This agent reads from and writes files to the local shared NFS file system. It connects with the agents to receive or send files to or from the NFS system.

BRKSAFE.DEFAULT is a WebSphere MQ File Transfer Edition agent on SYSD. It is defined by the broker when message flows with FTE nodes are deployed to the execution group *default*. It connects to the broker's local queue manager QMBRK via bindings mode. This agent receives and sends files from WebSphere Message Broker message flows via the FTE nodes.

WebSphere MQ Explorer and WebSphere MQ File Transfer Edition Explorer

The WebSphere MQ Explorer is used to view and administer WebSphere MQ queue managers and queue manager objects such as queues, topics, and channels. WebSphere MQ Explorer is built on an Eclipse integrated development environment, so it allows plug-ins to the base platform.

The WebSphere MQ File Transfer Edition Explorer is a plug-in to the WebSphere MQ Explorer. It is used to schedule file transfer requests and view the status of current requests. The tool includes a Transfer Log view that subscribes to the coordination queue manager for the audit information. The view displays information about every transfer that occurs in a given topology.

When we refer to the WebSphere MQ Explorer, we include the WebSphere MQ File Transfer Edition Explorer capabilities.

WebSphere Message Broker components

This section describes each WebSphere Message Broker component used to support this scenario.

WebSphere Message Broker Toolkit

The toolkit is used to create the message flows and artifacts deployed to the runtime broker BRKSAFE.

WebSphere Message Broker broker

BRKSAFE is the WebSphere Message Broker runtime broker. The broker is an ESB that routes messages, converts protocols, transforms data, and emits events. It has input and output nodes for a number of protocols, one of which is WebSphere MQ File Transfer Edition.

WebSphere Message Broker Explorer

The WebSphere Message Broker Explorer is a plug-in to the WebSphere MQ Explorer. It is used to administer the broker runtime environment. It shows the real-time status of brokers, execution groups, and message flows and their properties. With this plug-in, these components can be started and stopped using the WebSphere MQ Explorer. There are other ways to do this, such as the CLI and the configuration manager proxy API.

8.2.2 Scenario flow for inbound files

Figure 8-2 shows the flow for this scenario when files are inbound to the enterprise from the external partner.

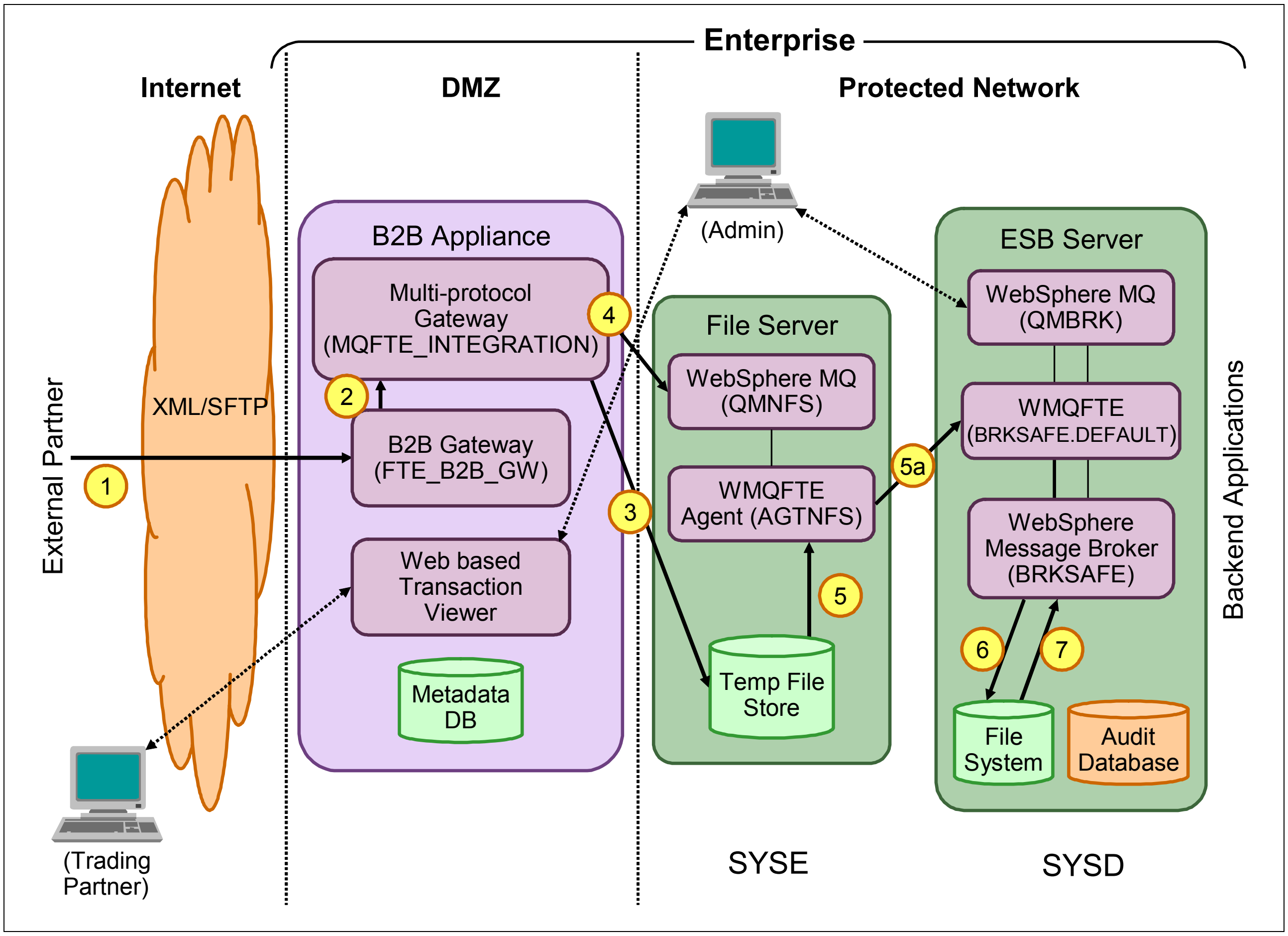


Figure 8-2 Inbound flow

Ports through the internal firewall need to be opened for connecting to the NFS shared file system and for putting MQ messages on queues hosted by QMNFS using the XB60 MQ client interface. The protected network will be protected by appropriate firewall rules.

Referring to Figure 8-2, the sequence of steps is as follows:

1. The external partner client initiates a file transfer using the SFTP protocol and sends the file to the FTE_B2B_GW B2B gateway in the Datapower XB60 B2B appliance.
2. The B2B gateway authenticates the client and validates that the client is a valid trading partner. It then passes the data via HTTP to the multi-protocol gateway, MQFTE_INTEGRATION, in the XB60.
3. The multi-protocol gateway service writes the transferred file to the shared Network File Server on SYSE, which resides within the protected network behind the internal firewall.

4. The multi-protocol gateway also connects to queue manager QMNFS using the XB60 MQ client and puts an XML transfer request message on the command queue of the AGTNFS agent. AGTNFS is running on the file server. The command message instructs AGTNFS to send the file stored on the NFS server to the broker agent.
5. AGTNFS reads the file from the temporary storage on the NFS file system.

   Then the agent sends the file over the MQ FTE backbone to the BRKSAFE.DEFAULT agent, which is running in a WebSphere Message Broker execution group named *default*.
6. The BRKSAFE.DEFAULT agent's properties are defined inside an FTEInput node in a message flow. The properties of the node define the directory to which the agent writes the file and the name of the file. When the transfer is complete, the message flow is initiated.
7. The FTEInput node of the message flow reads the file and parses it according to the domain defined in the FTEInput node. The BRKSAFE broker is the internal application in this scenario, but the file received is propagated from the broker to another back-end application via an MQOutput node, FTEOutput node, or anything supported by WebSphere Message Broker.

8.2.3 Scenario flow for outbound files

Figure 8-3 shows the flow for this scenario when files are outbound from the enterprise to the external partner.

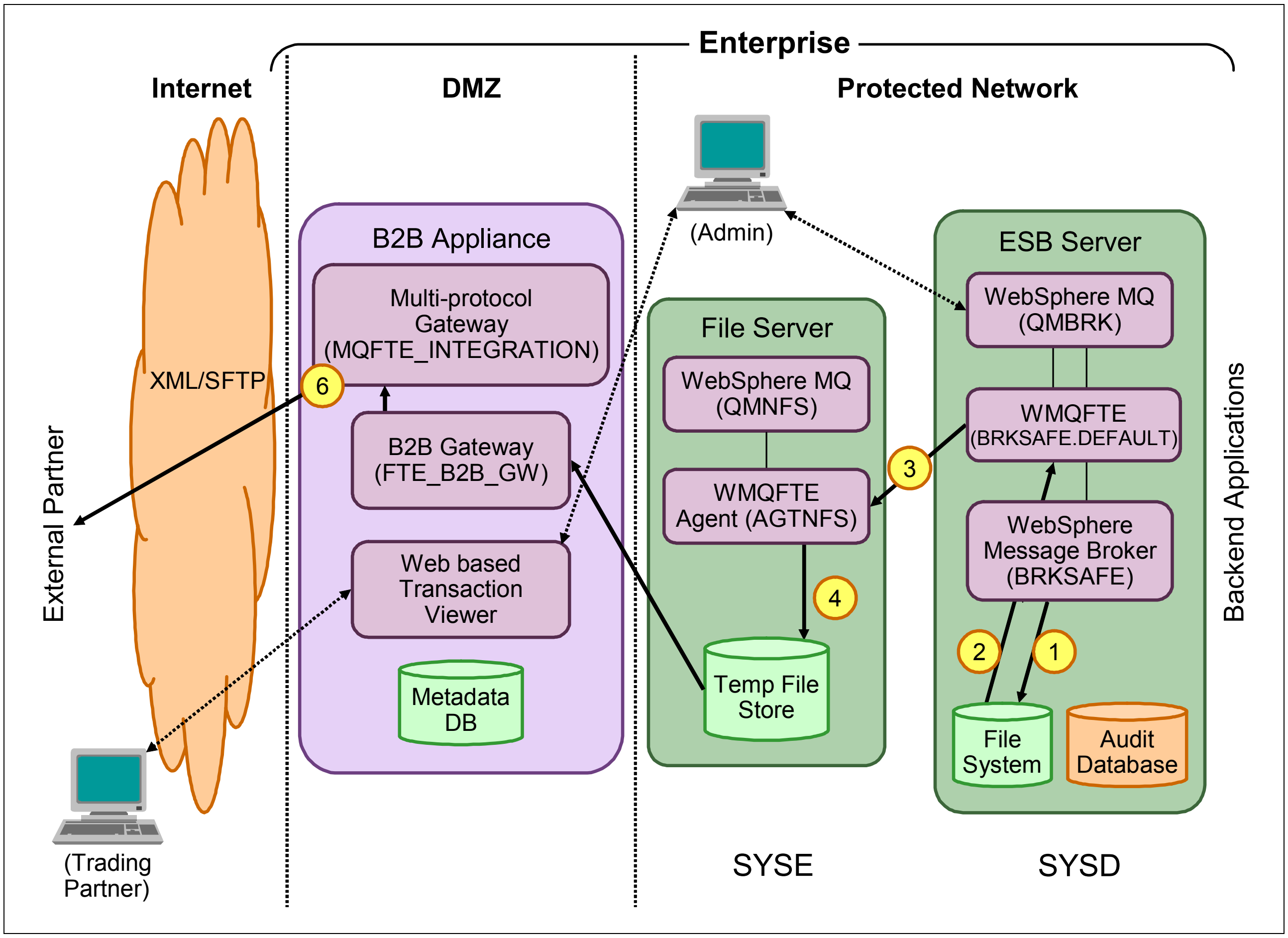


Figure 8-3 Outbound flow

Referring to Figure 8-3 on page 183, the sequence of steps is as follows:

1. A message flow in the broker is initiated by a back-end application putting a message on a queue that is defined on the flow's MQInput node or by writing a file that a FILEInput node is polling, or even sending a file over the MQ FTE backbone to the broker's agent for which a FTEInput node is waiting. Effectively, the flow can be initiated by any protocol supported by WebSphere Message Broker.

   In this scenario, the flow outputs an XML file using an FTEOuput node. The properties of this node define the agent that it will be using (BRKSAFE.DEFAULT, running in a WebSphere Message Broker execution group) and its queue manager (QMBRK).
2. The XML file is written to the file system by the message flow's FTEOutput node, invoking the transfer defined in the FTEOutput node.
3. The BRKSAFE.DEFAULT agent performs the transfer defined in the FTEOutput node properties—to the AGTNFS agent on SYSE, the NFS file server.
4. The AGTNFS agent writes the XML file to the local NFS file system that is shared with the XB60 appliance.
5. The B2B Gateway service, using an NFS poller front-side handler, polls the directory where AGTNFS has written the XML file. The file is parsed for sender and receiver information and validated against the profile repository, and attributes associated with the receiving partner profile are performed.
6. The B2B gateway routes the XML file to the receiving partner over SFTP based on the destination setting specified in the profile and gateway.

8.2.4 Protocols

In this scenario, the protocols in use between the external partner and the internal partner on the XB60 is SFTP. The protocols used during integration with the MQ FTE backbone are NFS for file integration and MQ for message-level integration. The MQ FTE backbone utilizes both MQ and file system protocols to move files from one location to the next.

WebSphere Message Broker uses WebSphere MQ messaging and WebSphere MQ File Transfer Edition agents driven by FTEInput and FTEOutput nodes in the message flows.

8.2.5 Security

In this scenario, we are concerned with securing data during transmission over SFTP and the MQ FTE backbone. Authentication in the B2B gateway is handled by the XB60.

Transport security

The protocol used between the external partner and the XB60 B2B gateway is SFTP. SFTP refers to the SSH File Transfer Protocol. It is a network protocol that provides file transfer over any reliable data stream. This protocol assumes that it is run over a secure channel such as SSH, that the server has already authenticated the client, and that the identity of the client user is available to the server.

WebSphere MQ File Transfer Edition security

For any file transfer request, the agent processes require a certain level of access to its local file systems. In addition, both the user identifier associated with the agent process and the user identifiers associated with users performing file transfer operations must have the authority to use certain WebSphere MQ objects. Because the BRKSAFE.DEFAULT agent is running in the broker's execution group, it assumes the identity of the user running the broker process on SYSD.

Commands are issued by users, who might be in operational roles in which they typically start file transfers. Alternatively, they might be in administrative roles, in which they can additionally control when agents are created, started, deleted, or cleaned (that is, when messages from all agent system queues are removed). Messages containing command requests are placed on an agent's SYSTEM.FTE.COMMAND queue when a user issues a command. The agent process retrieves messages containing command requests from the SYSTEM.FTE.COMMAND queue. The agent process also uses four other system queues, which are:

- SYSTEM.FTE.DATA.*agent_name*
- SYSTEM.FTE.EVENT.*agent_name*
- SYSTEM.FTE.REPLY.*agent_name*
- SYSTEM.FTE.STATE.*agent_name*

WebSphere MQ File Transfer Edition supports finer-grained checking of users' authorities, which permits access to be granted (or denied) to specific product functions for each user. For example, you can choose which users have the authority to schedule transfer operations to happen at a future time. Because users issuing commands use the above queues in different ways from the agent process, assign different WebSphere MQ authorities to the user identifiers or user groups associated with each. For more information, see *Using groups to manage authorities for resources specific to WebSphere File Transfer Edition* at:

http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/topic/com.ibm.wmqfte.admin.doc/group_resource_access.htm

The agent process has additional queues that can be used to grant users the authority to perform certain actions. The agent does not put or get messages on these queues. However, you must ensure that the queues are assigned the correct WebSphere MQ authorities, both for the user identifier used to run the agent process and for the user identifiers associated with users who are being authorized to perform certain actions. The authority queues are:

- SYSTEM.FTE.AUTHADM1.*agent_name*
- SYSTEM.FTE.AUTHAGT1.*agent_name*
- SYSTEM.FTE.AUTHMON1.*agent_name*
- SYSTEM.FTE.AUTHOPS1.*agent_name*
- SYSTEM.FTE.AUTHSCH1.*agent_name*
- SYSTEM.FTE.AUTHTRN1.*agent_name*

The agent process also publishes messages to the SYSTEM.FTE topic on the coordination queue manager using the SYSTEM.FTE queue. Depending on whether the agent process is in the role of the source agent or the destination agent, the agent process might require authority to read, write, update, and delete files.

You can create and modify authority records for WebSphere MQ objects using the WebSphere MQ Explorer. Right-click the object and then click **Object Authorities** → **Manage Authority Records**. You can also create authority records using the **`setmqaut`** command.

Instead of granting authority to individual users for all of the various objects that might be involved, configure two security groups for the purposes of administering WebSphere MQ File Transfer Edition access control:

- FTEUSER
- FTEAGENT

Firewall security

Firewall configuration plays an important role in securing your connections to and from external partners and in protecting the internal network.

The external firewall must allow incoming requests from all of the trading partner's source IP addresses or range of IP addresses. This can be configured in the firewall rules configuration. The method used to configure firewall rules depends on the model and type of firewall being used.

The demilitarized zone (DMZ) is a termination point at the edge of the protected network and typically is used to house internet-facing systems.

Setting up tight rules on the inner firewall is important for protecting your internal systems. Typically, inner firewall rules are set up to allow only traffic from a mediation server in the DMZ that terminates the connection from the internet. The mediation server then re-establishes the connection through the inner firewall to a system the files are destined for or to a system that moves the files to a back-end application.

For this scenario we need to open a TCP port for MQ and multiple ports for NFS (TCP and UDP 2049 and 111).

8.2.6 Software and hardware prerequisites

This scenario solution uses the following software:

- WebSphere MQ V7.0.1
- WebSphere Message Broker V7.0.0.1
- WebSphere MQ File Transfer Edition V7.0.2
- WebSphere DataPower B2B Appliance Model 9235-62X with firmware Version 3.8.1.2 or later

8.3 Configuring WebSphere MQ File Transfer Edition

For this scenario, we assume that the WebSphere MQ File Transfer Edition configuration is in place. To prepare for the scenario, simply start AGTNFS.

8.3.1 Security considerations

When trading files with external partners through a B2B gateway product such as the WebSphere DataPower B2B Appliance XB60, all of the security and governance needed between the external partner and the business-to-business hub is provided in the configuration and the protocols that you use to exchange files. The partners do not know anything about or have any access to the WebSphere MQ File Transfer Edition queue managers and agents. To WebSphere MQ File Transfer Edition, the XB60 is the user. A single agent can be used and partners can be identified in the metadata that the XB60 sends to the agent. For securing internal file transfers, WebSphere MQ File Transfer Edition utilizes MQ Security and has added security to control WebSphere MQ File Transfer Edition functionality. However, for this scenario we do not implement WebSphere MQ File Transfer Edition security.

8.3.2 Configuration prerequisites

AGTNFS is used to integrate to the XB60. Its purpose for the inbound flow is to pick files up from the shared NFS mount point (`/XB60/ftewmb/In`) and transfer them to any other agent and directory that is specified in the XML command file used by the XB60 to trigger the file transfer. The agent's purpose for the outbound flow is to receive a file from any other agent and to write that file to the shared NFS mount point (`/XB60/ftewmb/Out`) that is being polled by the XB60 for XML files destined for external partners.

This scenario assumes that the following configuration has been created for the WebSphere MQ File Transfer Edition and WebSphere Message Broker components.

- To prepare the file systems, the `/ftewmb/In` and `/ftewmb/out` directories must be created in the `/XB60` directory on the system running the NFS share. Ensure that permissions allow read, write, and execute.
- The following queue managers and listener ports must be configured in WebSphere MQ:
  - QMNFS (1415) on SYSE
  - QMBRK (14015) on SYSD
  - QMSAFE (14014) on SYSD

  For information about how to create queue managers, see Appendix A, "Configuration of WebSphere MQ File Transfer Edition" on page 269.
- The AGTNFS WebSphere MQ File Transfer Edition agent must be created. For information about how to create this agent, see Appendix A, "Configuration of WebSphere MQ File Transfer Edition" on page 269.

> **BRKSAFE.DEFAULT creation:** BRKSAFE.DEFAULT is created by WebSphere Message Broker and is not part of the normal configuration of WebSphere MQ File Transfer Edition agents.

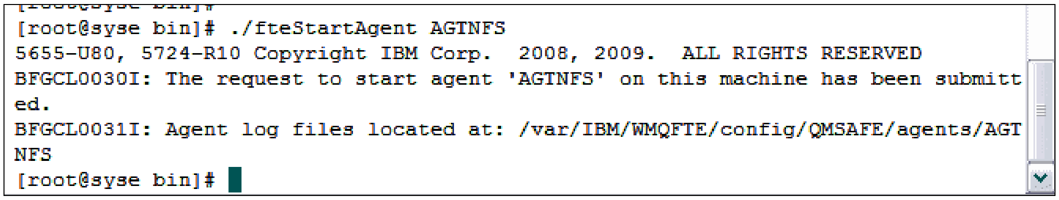
8.3.3 Starting AGTNFS

For this scenario AGTNFS and the NFS services are running on a Linux system.

1. To start AGTNFS, log onto the computer running the agent that is sharing the NFS mount point with the XB60.
2. Navigate to the WebSphere MQ File Transfer Edition bin directory, and at the prompt enter the following command:

   ```
   ./fteStartAgent AGTNFS
   ```

   Figure 8-4 illustrates the response upon a successful start of the agent.



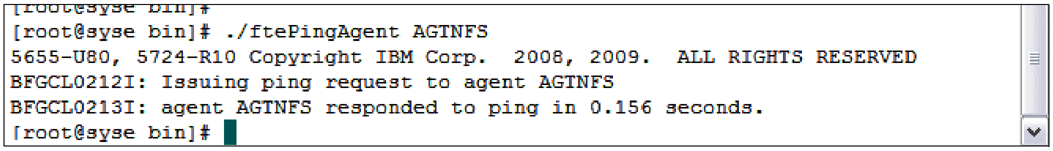
```
[root@syse bin]# ./fteStartAgent AGTNFS
5655-U80, 5724-R10 Copyright IBM Corp.  2008, 2009.  ALL RIGHTS RESERVED
BFGCL0030I: The request to start agent 'AGTNFS' on this machine has been submitt
ed.
BFGCL0031I: Agent log files located at: /var/IBM/WMQFTE/config/QMSAFE/agents/AGT
NFS
[root@syse bin]#
```

Figure 8-4 Start AGTNFS

3. Ping AGTNFS to be sure that it is running with the following command:

   ```
   ./ftePingAgent AGTNFS
   ```

Figure 8-5 illustrates the response upon a successful ping.



```
[root@syse bin]#
[root@syse bin]# ./ftePingAgent AGTNFS
5655-U80, 5724-R10 Copyright IBM Corp.  2008, 2009.  ALL RIGHTS RESERVED
BFGCL0212I: Issuing ping request to agent AGTNFS
BFGCL0213I: agent AGTNFS responded to ping in 0.156 seconds.
[root@syse bin]#
```

Figure 8-5 Ping AGTNFS

8.4 WebSphere DataPower B2B Appliance XB60 configuration

This section describes the configuration steps required in the XB60 to implement the scenario flows as depicted in Figure 8-2 on page 182 and Figure 8-3 on page 183.

- 8.4.1, “Configuration prerequisites” on page 188
- 8.4.2, “Configuring a static NFS mount object” on page 188
- 8.4.3, “Configuring an MQ Queue Manager object” on page 190
- 8.4.4, “Configuring a multi-protocol gateway for MQFTE integration” on page 193
- 8.4.5, “Configuring trading partner profiles” on page 203
- 8.4.6, “Configuring the B2B gateway” on page 210

The installation of the XB60 is not covered in this book. For information about how to install the appliance, see the installation documentation.

For the purpose of this scenario, we configure a B2B gateway in a single domain that acts as your business-to-business hub. The partner's business-to-business hub is preconfigured in its own domain to allow us to simulate an external partner without the need for another machine.

8.4.1 Configuration prerequisites

This scenario assumes that the following configuration is in place:

- An NFS mount point must be exported on a file server system, access must be granted for the XB60’s IP address, and file and directory permissions must be set to allow read, write, and execute access.
- The `C:/FTEin`, `C:/FTEout`, and `C:/FTEtrace` directories must be created on the target system.
- You must have valid user credentials on the XB60.
- On the XB60 an application domain other then default must be created for this scenario.

8.4.2 Configuring a static NFS mount object

The static NFS mount object is needed to support the connection to the NFS directory that is shared between AGTNFS and the XB60. This mount point is used by the multi-protocol gateway policy being used for WebSphere MQ File Transfer Edition integration.

To configure the Static NFS Mount object use the following the steps:

1. From the left navigation menu of the XB60 click **Objects**, and then navigate to Network Settings and click **NFS Static Mounts** (Figure 8-6).

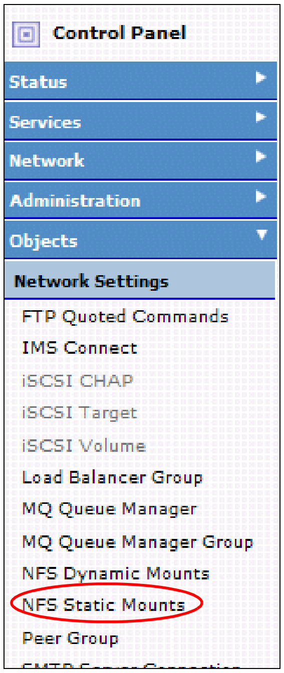
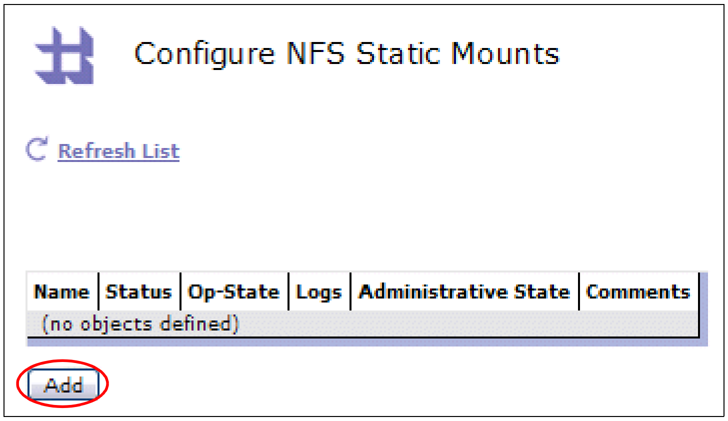


Figure 8-6 Left navigation menu - NFS

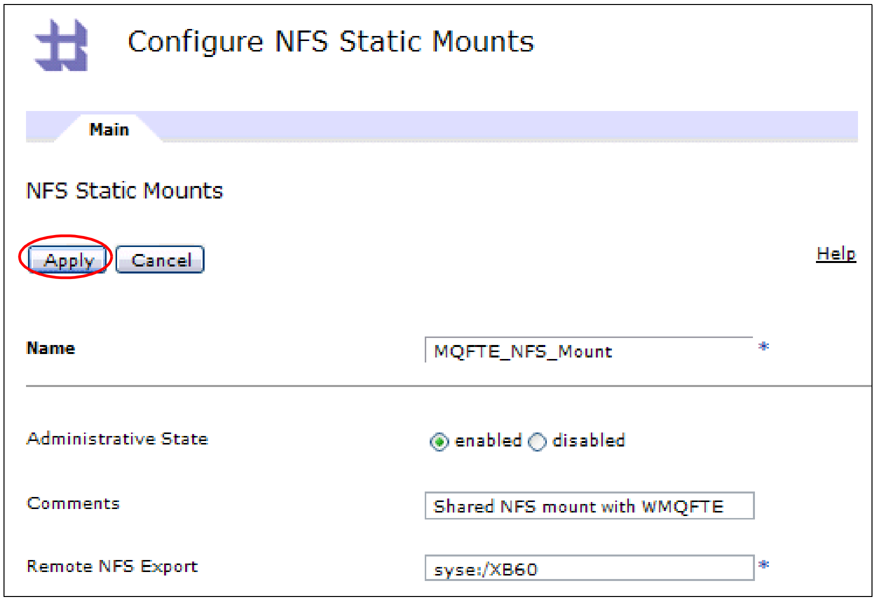
2. In the Configure NFS Static Mounts list view, click **Add** (Figure 8-7).



Name	Status	Op-State	Logs	Administrative State	Comments
(no objects defined)					

Figure 8-7 Configure NFS Static Mounts list view

Figure 8-8 shows the Main tab of the Configure NFS Mounts page.



NFS Static Mounts

Name: MQFTE_NFS_Mount *

Administrative State: ● enabled ○ disabled

Comments: Shared NFS mount with WMQFTE

Remote NFS Export: syse:/XB60 *

Figure 8-8 NFS Static Mount object

Complete these fields as follows:

a. Enter `MQFTE_NFS_Mount` in the Name field.

b. Make sure that the administrative state is **enabled**.

c. In the Remote NFS Export field enter the IP address or host name of the file server followed by the directory path. For the purpose of this scenario use `syse:/XB60`.

d. Leave all remaining fields with the default values and click **Apply**.

3. Save the configuration by clicking **Save Config** in the upper right corner of the page.

> **Troubleshooting tip:** You might see that the object is down because the process is carried out asynchronously. Click **Configure NFS Static Mounts** at the top of this view and you are taken back to the list view, where you should see that the mount is now up. If the object still does not come up, you might have the wrong values in the Remote NFS Export field, or the host or NFS is not accessible. Check to be sure that the NFS Mount is up, the directory rights are set correctly, and the host is accessible from the XB60. You can ping the host in the DataPower Troubleshooting page, which is accessible from the Control Panel.

8.4.3 Configuring an MQ Queue Manager object

The MQ Queue Manager object is needed to support the connection to the command queue that consumes the XML command message that triggers the file transfer between AGTNFS and BRKSAFE.DEFAULT. This queue manager information is used by the multi-protocol gateway processing policy being used for WebSphere MQ File Transfer Edition integration.

To configure the MQ Queue Manager object for QMNFS use the following the steps:

1. From the left navigation menu click **Network**, and then navigate to **Other** and click **MQ Queue Manager** (Figure 8-9).

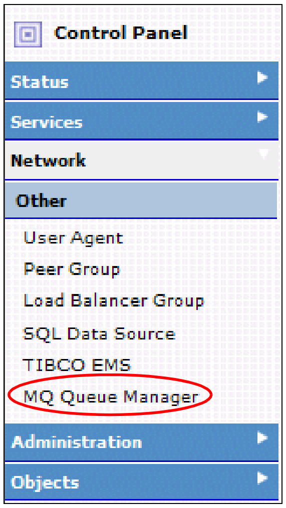
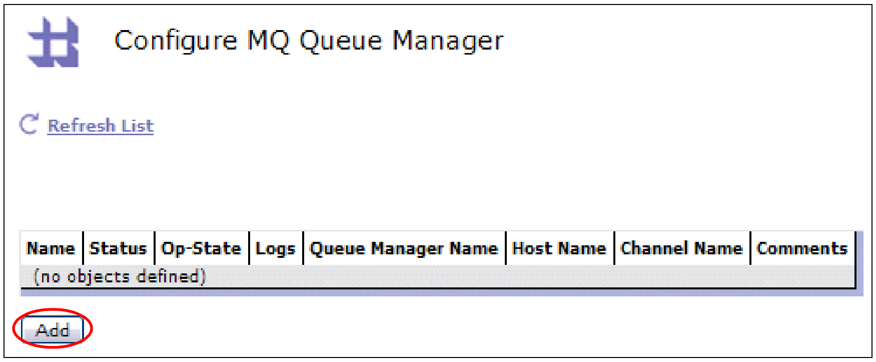


Figure 8-9 Left Navigation Menu - Queue Manager

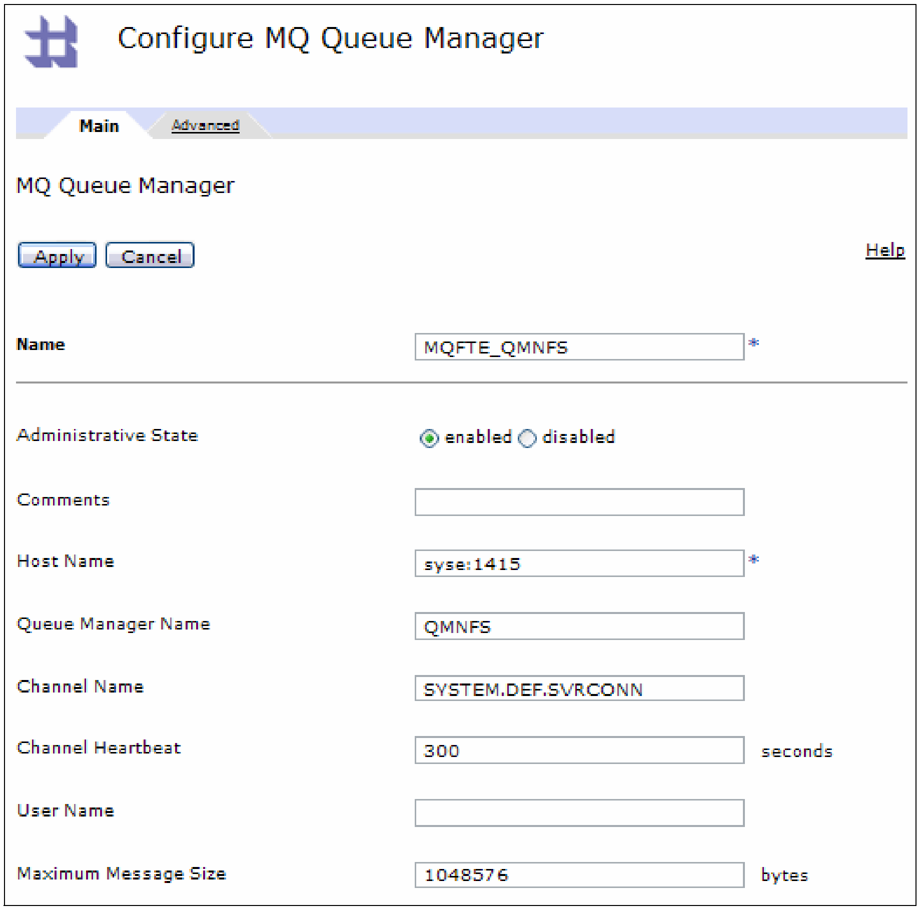
2. In the Configure MQ Queue Manager list view, click **Add** (Figure 8-10).



Name	Status	Op-State	Logs	Queue Manager Name	Host Name	Channel Name	Comments
(no objects defined)							

Figure 8-10 Configure Queue Manager list view

3. In the Configure MQ Queue Manager view the first tab is the Main tab, and it is the only tab that we configure. Figure 8-11 illustrates how the configured MQ Queue Manager object should look. Fields that are populated with default values might not be displayed.



Configure MQ Queue Manager

MQ Queue Manager

Name	MQFTE_QMNFS *
Administrative State	● enabled ○ disabled
Comments	
Host Name	syse:1415 *
Queue Manager Name	QMNFS
Channel Name	SYSTEM.DEF.SVRCONN
Channel Heartbeat	300 seconds
User Name	
Maximum Message Size	1048576 bytes

Figure 8-11 MQ Queue Manager object

Complete these fields as follows:

a. On the Main tab enter `MQFTE_QMNFS` in the Name field and make sure that the administrative state is **enabled**.

b. In the Host Name field enter the IP address or host name of the server running the queue manager followed by the listener port number for the queue manager (in this case, `syse:1415`).

c. In the Queue Manager Name field type `QMNFS`.

d. Leave all remaining fields with the default values and click **Apply**.

4. Save the configuration by clicking **Save Config** in the upper right corner of the page.

> **Troubleshooting tip:** You might see that the object is down because the process is carried out asynchronously. Click **Configure MQ Queue Manager** at the top of this view and you are taken back to the list view, where you should see that the mount is now up. If the object still does not come up, you might have the wrong values in the Host Name, Port, or Queue manager fields, or the host or MQ is not accessible. Check to be sure that the queue manager is up and that the host is accessible from the XB60. You can ping the host in the DataPower Troubleshooting page, which is accessible from the Control Panel.

8.4.4 Configuring a multi-protocol gateway for MQFTE integration

The multi-protocol gateway is used to integrate to MQ File Transfer Edition through a shared NFS mount point. The gateway uses a custom action to write the file to the file system with a unique file name and to generate and send an XML command message that triggers the file transfer.

1. From the left navigation menu click **Services** → **New Multi-Protocol Gateway** (Figure 8-12).

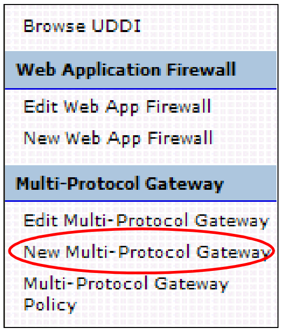
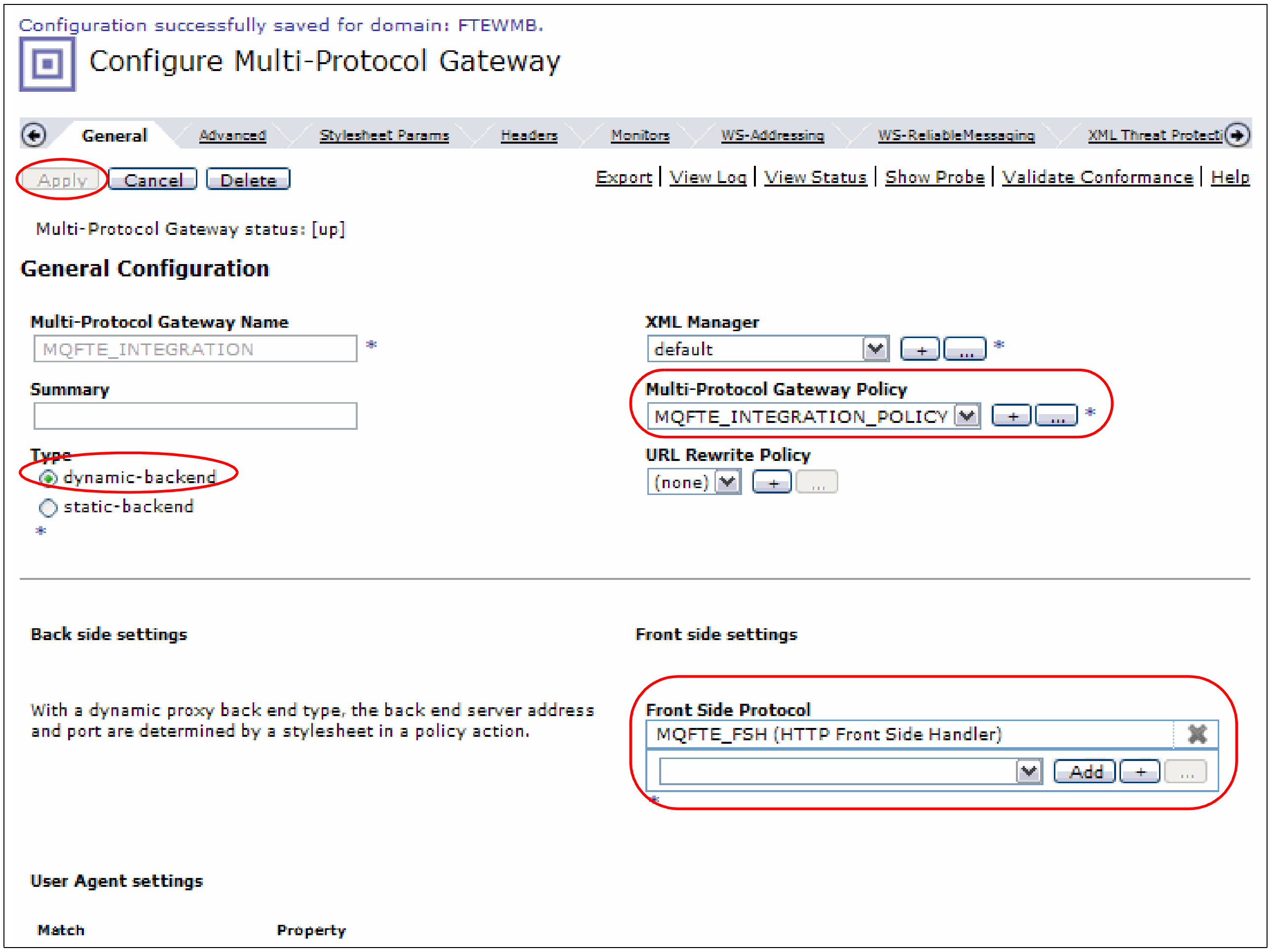


Figure 8-12 Left Navigation Menu - Multi-Protocol Gateway

2. In the Configure Multi-Protocol Gateway view the first tab is the General tab, and it is the only tab that we configure. This scenario does not require the use of the other tabs (Figure 8-13).



Configuration successfully saved for domain: FTEWMB.

Configure Multi-Protocol Gateway

Multi-Protocol Gateway status: [up]

General Configuration

Multi-Protocol Gateway Name: MQFTE_INTEGRATION *

Summary:

Type
● dynamic-backend
○ static-backend
*

XML Manager: default *

Multi-Protocol Gateway Policy: MQFTE_INTEGRATION_POLICY *

URL Rewrite Policy: (none)

Back side settings

With a dynamic proxy back end type, the back end server address and port are determined by a stylesheet in a policy action.

Front side settings

Front Side Protocol: MQFTE_FSH (HTTP Front Side Handler)

User Agent settings

Figure 8-13 Configure multi-protocol gateway

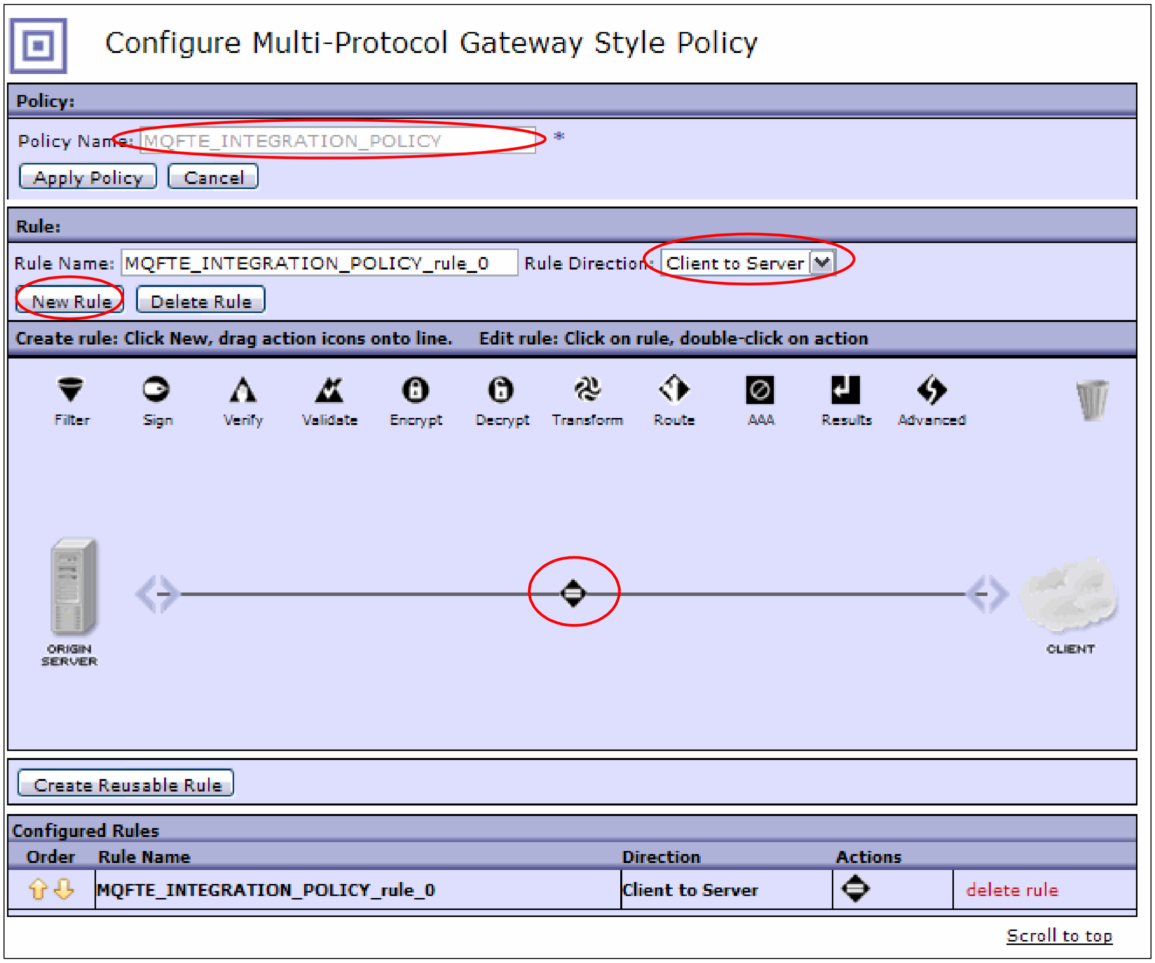
Complete these fields as follows:

a. In the Multi-Protocol Gateway Name field enter `MQFTE_INTEGRATION`.

b. In the Type field select **dynamic-backend**.

c. Set the Request Type and Response Type fields to **Non-XML**. (These are located lower on the page and are not shown in Figure 8-13.)

3. Create a multi-protocol gateway policy by clicking the Plus icon next to the Multi-Protocol Gateway Policy drop-down menu (Figure 8-13 on page 194). This opens the Configure Multi-Protocol Gateway Policy view (Figure 8-14).



Configure Multi-Protocol Gateway Style Policy

Policy:

Policy Name: MQFTE_INTEGRATION_POLICY *

Apply Policy | Cancel

Rule:

Rule Name: MQFTE_INTEGRATION_POLICY_rule_0 Rule Direction: Client to Server

New Rule | Delete Rule

Create rule: Click New, drag action icons onto line. Edit rule: Click on rule, double-click on action

Create Reusable Rule

Configured Rules

Order	Rule Name	Direction	Actions	
⇧⇩	MQFTE_INTEGRATION_POLICY_rule_0	Client to Server	◆	delete rule

Scroll to top

Figure 8-14 Configure multi-protocol gateway policy

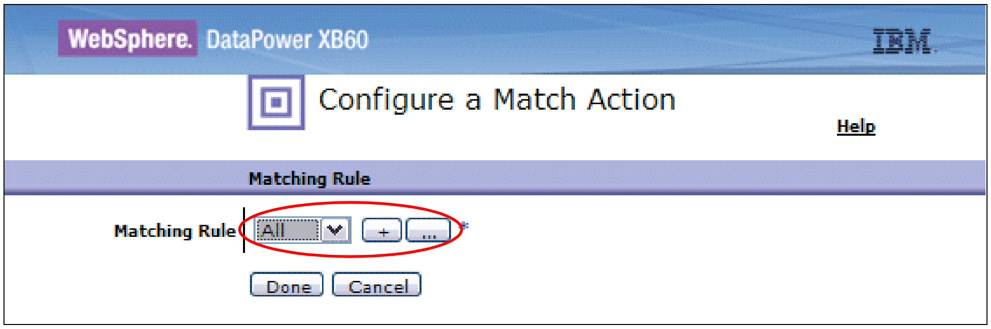
Complete these fields as follows:

a. Enter `MQFTE_INEGRATION_POLICY` in the Policy Name field.

b. Click **New Rule** to create the first rule. The rule name is filled in for you and the Match action (◆) is automatically added to the pallet (Figure 8-14).

c. Change the Rule Direction to **Client to Server**.

d. Double-click the Match action to configure it. The Configure a Match Action view displays (Figure 8-15).



Configure a Match Action

Matching Rule

Matching Rule: All

Figure 8-15 Configure Match Action view

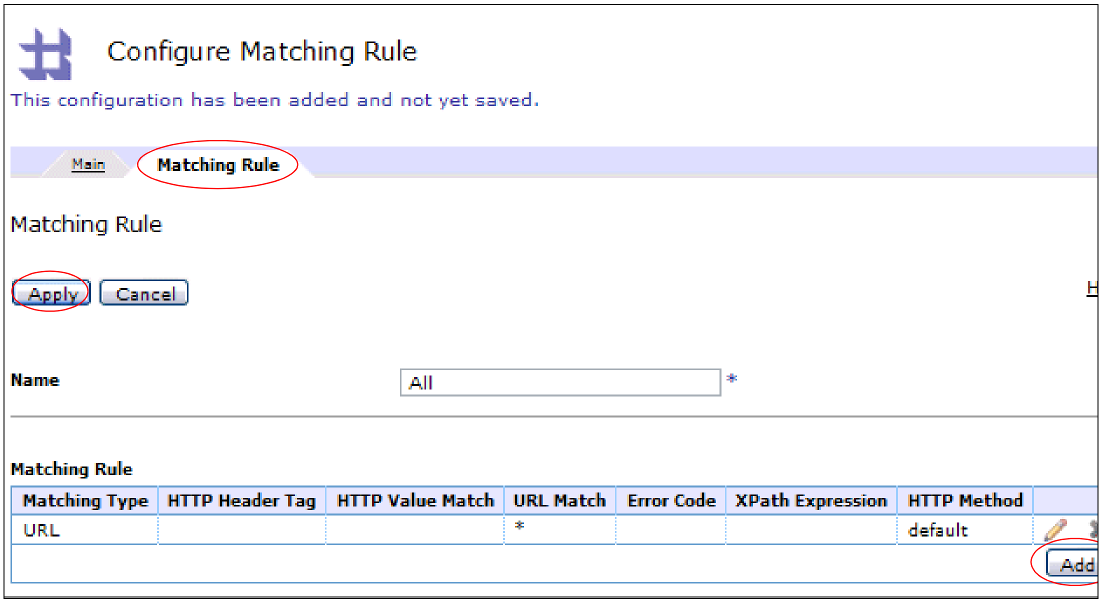
Complete these fields as follows:

i. Use the drop-down menu in the Matching Rule field to select **All**.

> **Attention:** If All is not displayed in the drop-down menu it must be created by following steps ii through viii. If All is displayed select it and skip to step ix.

ii. Click the Plus icon next to the Matching Rule field.

iii. Click the **Matching Rule** tab and type `All` in the Name field (Figure 8-16).



Configure Matching Rule

This configuration has been added and not yet saved.

Matching Rule

Name: All

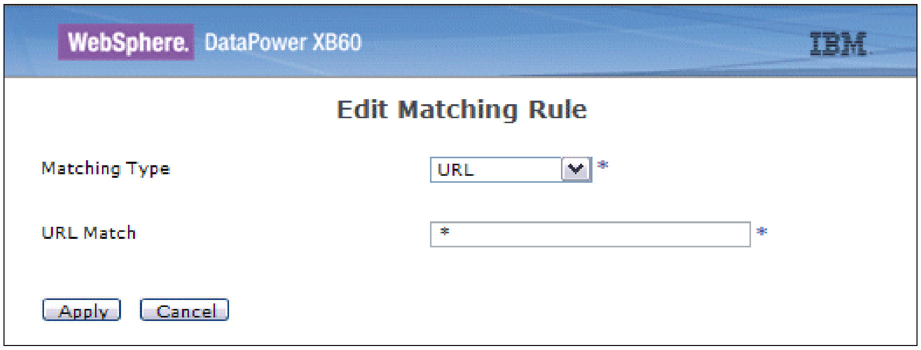
Matching Rule

Matching Type	HTTP Header Tag	HTTP Value Match	URL Match	Error Code	XPath Expression	HTTP Method
URL			*			default

Figure 8-16 Configure Matching Rule

iv. Click **Add** to add a matching rule.

v. In the Edit Matching Rule panel (Figure 8-17), select **URL** in the Matching Type field and type * in the URL Match field.



Edit Matching Rule

Matching Type: URL *

URL Match: * *

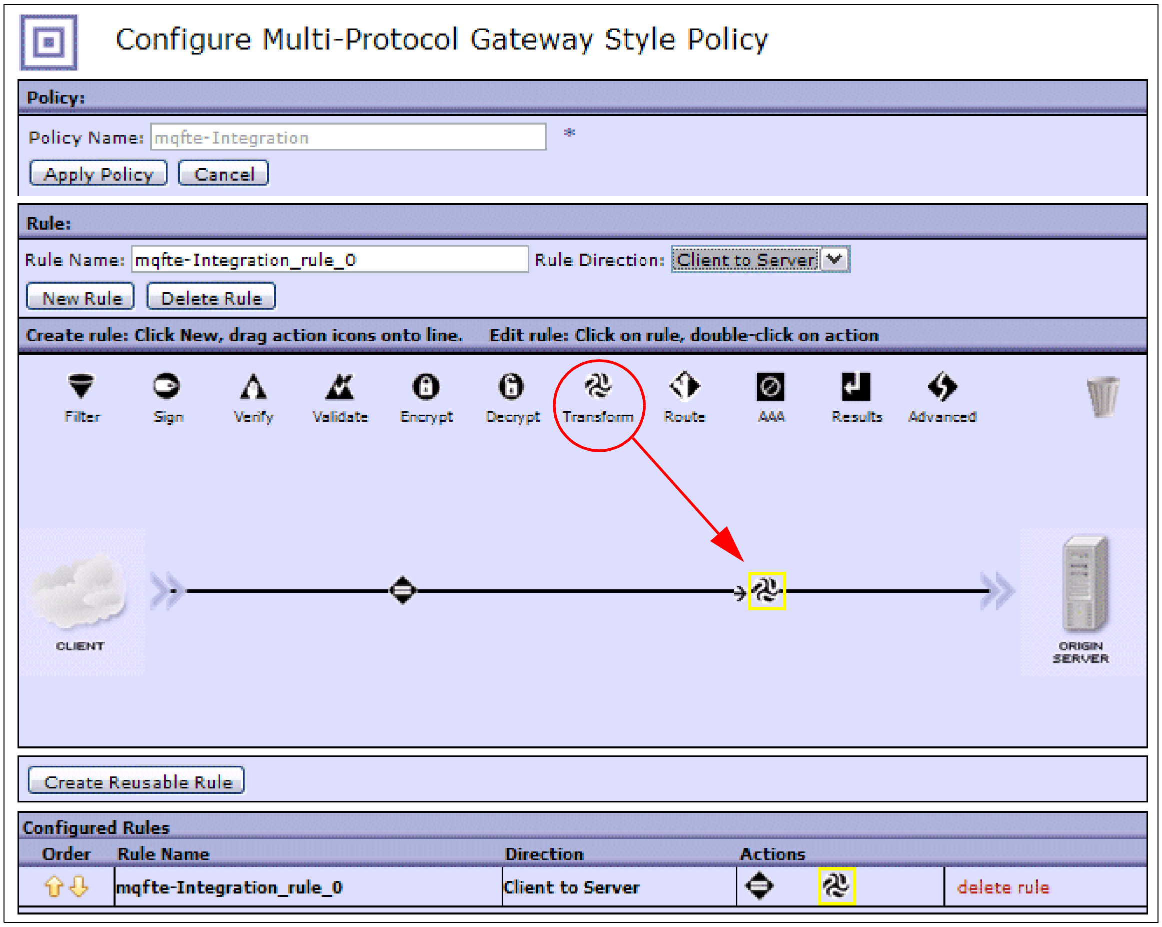
Figure 8-17 Configure Matching Rule

vi. Click **Apply** in the Edit Matching Rule view.

vii. Click **Apply** in the Configure Matching Rule view.

viii.Click **Done** in the Configure a Match Action view (Figure 8-15 on page 196) and you are returned to the Configure Multi-Protocol Gateway Style Policy view.

e. Drag a **Transform** action onto the palette to the right of the Match action (Figure 8-18).



Configure Multi-Protocol Gateway Style Policy

Policy:

Policy Name: mqfte-Integration *

Rule:

Rule Name: mqfte-Integration_rule_0 Rule Direction: Client to Server

Create rule: Click New, drag action icons onto line. Edit rule: Click on rule, double-click on action

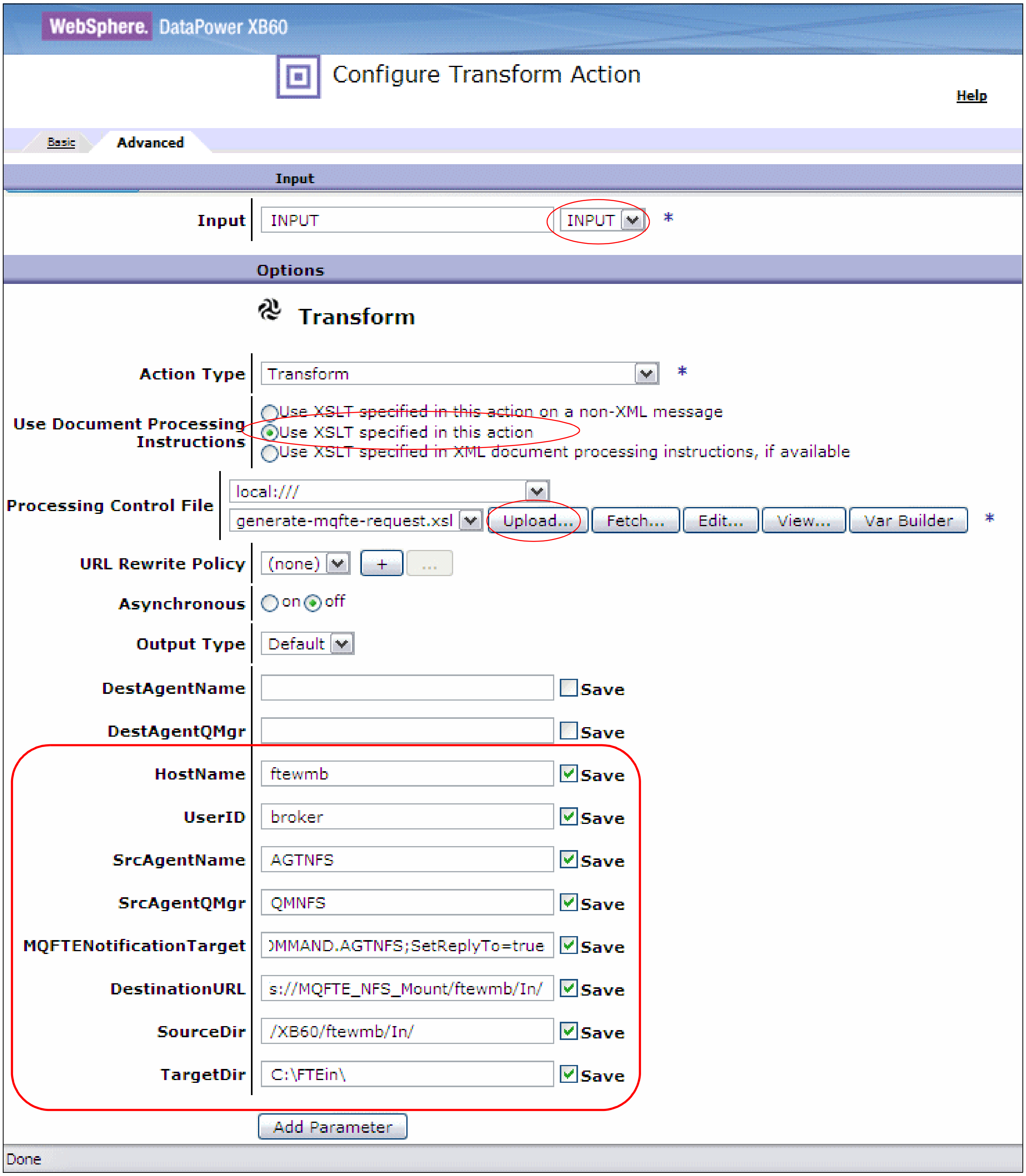
Configured Rules

Order	Rule Name	Direction	Actions	
⇧⇩	mqfte-Integration_rule_0	Client to Server		delete rule

Figure 8-18 Multi-protocol gateway style policy - Transform action

f. Double-click **Transform Action** to configure it.

Figure 8-19 illustrates how the configured Transform action should look after you complete the configuration. Fields that are populated with default values might not be displayed.



Input

Input	INPUT INPUT *

Options

Transform

Action Type	Transform *
Use Document Processing Instructions	○ Use XSLT specified in this action on a non-XML message ● Use XSLT specified in this action ○ Use XSLT specified in XML document processing instructions, if available
Processing Control File	local:/// generate-mqfte-request.xsl Upload... Fetch... Edit... View... Var Builder *
URL Rewrite Policy	(none) + ...
Asynchronous	○ on ● off
Output Type	Default
DestAgentName	□ Save
DestAgentQMgr	□ Save
HostName	ftewmb ☑ Save
UserID	broker ☑ Save
SrcAgentName	AGTNFS ☑ Save
SrcAgentQMgr	QMNFS ☑ Save
MQFTENotificationTarget	)MMAND.AGTNFS;SetReplyTo=true ☑ Save
DestinationURL	s://MQFTE_NFS_Mount/ftewmb/In/ ☑ Save
SourceDir	/XB60/ftewmb/In/ ☑ Save
TargetDir	C:\FTEin\ ☑ Save

Figure 8-19 Configure Transform action

This Transform action utilizes the generate-mqfte-request.xsl stylesheet to write files to the NFS file share location (defined in the DestinationURL and SourceDir parameters) with a unique file name. It uses the HostName, SrcAgentName, SrcAgentQMgr, TargetDir, and UserID parameters to generate the XML transfer command message used to trigger the file transfer in the MQ FTE backbone. It sends the XML command message to the SYSTEM.FTE.COMMAND.AGTNFS queue (defined in the MQFTENotificationTarget parameter) and returns an NFS and MQ response back to the result rule in the policy.

The XSL also takes input from the URL string received from the internal profile and sets the TgtAgentName and TgtAgentQMgr in the XML command message.

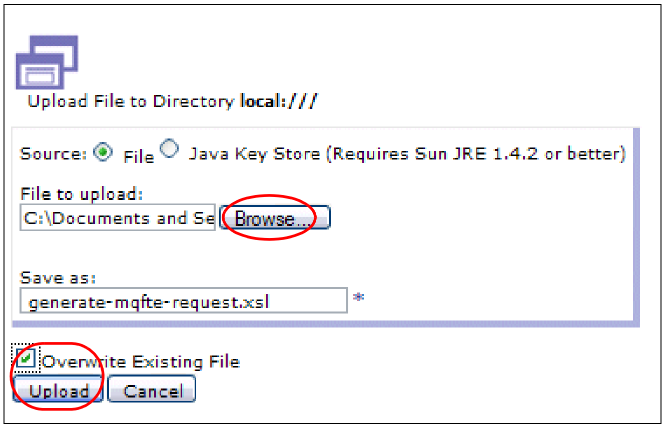
i. Click the **Advanced** tab and select **INPUT** from the drop-down menu for the Input field.

ii. Be sure that **Use XSLT specified in this action** is selected in the Use Document Processing Instructions field and click **Upload** to upload the XSL that we use for this policy.

iii. Click **Upload**.

iv. In the Upload File view (Figure 8-20) click **Browse**, navigate to the XSL file, and click **Open**.

> **Additional material:** The `generate-mqfte-request.xsl` file, located in the `Common_Files` directory, can be uploaded for use as the XSL file.



Upload File to Directory **local:///**

Source: ● File ○ Java Key Store (Requires Sun JRE 1.4.2 or better)

File to upload: C:\Documents and Se

Save as: generate-mqfte-request.xsl *

☑ Overwrite Existing File

Figure 8-20 Upload XSL File

Be sure that **Overwrite Existing File** is selected, then click **Upload**. Click **Continue** in the confirmation box to return to the Advanced tab.

v. Back in the Configure Transform Action view (Figure 8-19 on page 198), configure the parameters used in the `generate-mqfte-request.xsl` processing control file using Table 8-1.

Table 8-1 generate-mqfte-request.xsl parameters

Parameter name	**Value**
HostName	`ftewmb`
UserID	`broker`
SrcAgentName	`AGTNFS`

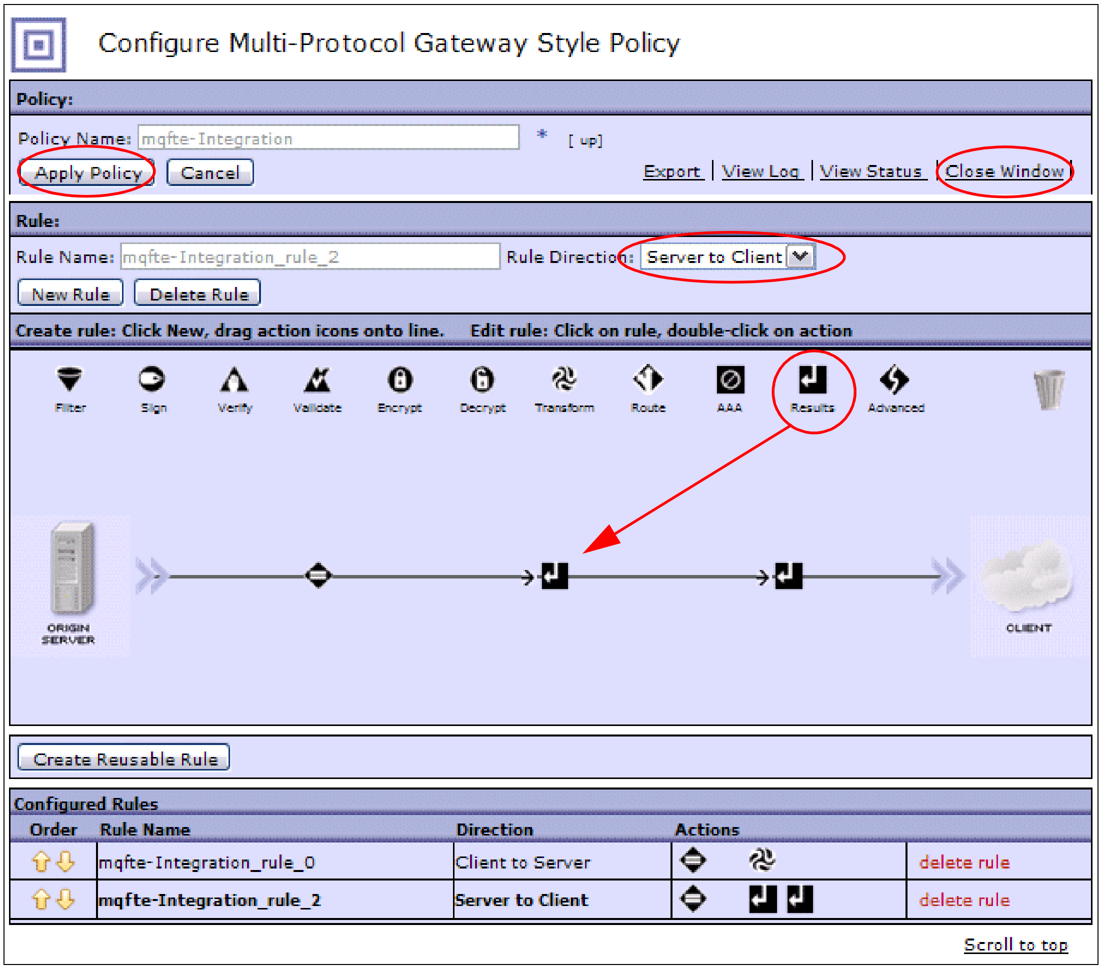
Parameter name	Value
SrcAgentQMgr	`QMNFS`
MQFTENotificationTarget	`dpmq://MQFTE_QMNFS/syse:1415?RequestQueue=SYSTEM.FTE.COMMAND.AGTNFS;SetReplyTo=true`
DestinationURL	`dpnfs://MQFTE_NFS_Mount/ftewmb/In/`
SourceDir	`/XB60/ftewmb/In/`
TargetDir	`C:\FTEin\`

vi. Click the drop-down menu for the Output field and select **OUTPUT**.

vii. Leave the default values in all other fields and click **Done** to complete the configuration of the Transform action, which returns you to the Configure Multi-Protocol Gateway Style Policy view.

g. Click **New Rule** to create the second rule. This rule is used for the response. The rule name is filled in for you and the Match action is automatically added to the pallet.

h. In the Configure Multi-Protocol Gateway Style Policy view (Figure 8-21), change the rule direction to **Server to Client**.



Configure Multi-Protocol Gateway Style Policy

Policy:

Policy Name: mqfte-Integration * [up]

Rule:

Rule Name: mqfte-Integration_rule_2 Rule Direction: Server to Client

Create rule: Click New, drag action icons onto line. Edit rule: Click on rule, double-click on action

Configured Rules

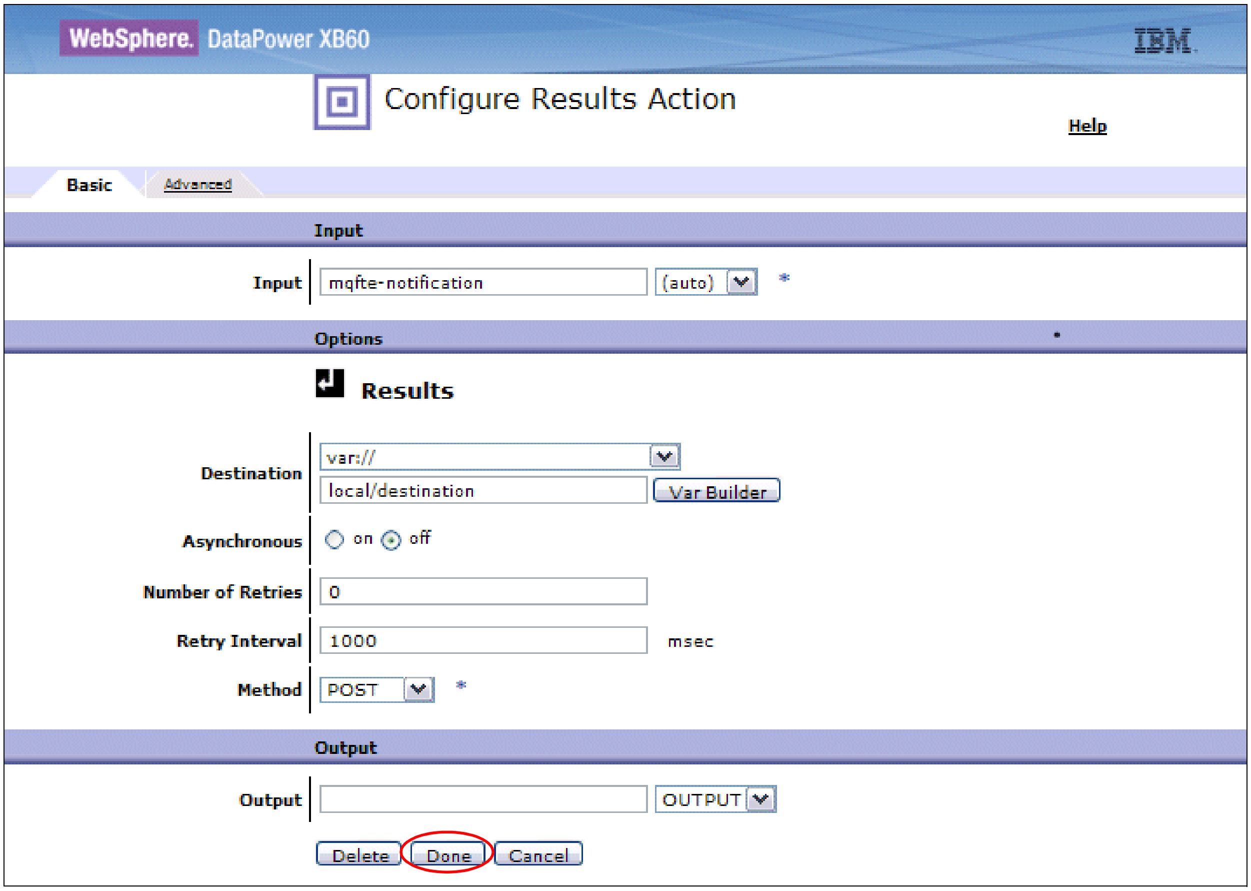
Order	Rule Name	Direction	Actions	
⇧⇩	mqfte-Integration_rule_0	Client to Server		delete rule
⇧⇩	**mqfte-Integration_rule_2**	**Server to Client**		delete rule

Figure 8-21 Multi-protocol gateway style policy - Configure Results action

i. Double-click the Match action (◆) and use the drop-down menu to set it to **All**, and then click **Done**.

j. Drag a **Results** action onto the palette to the right of the Match action. Then double-click the **Results** action to configure it.

Figure 8-22 illustrates how the configured Results action should look when its configuration is complete. Fields that are populated with default values might not be displayed.



Configure Results Action

Input

Input: mqfte-notification (auto) *

Options

Results

Destination: var://
local/destination

Asynchronous: ○ on ● off

Number of Retries: 0

Retry Interval: 1000 msec

Method: POST *

Output

Output: OUTPUT

Figure 8-22 Configure Results action

Complete these fields as follows:

i. On the Basic tab type `mqfte-notification` in the Input field.

ii. In the Destination field click the drop-down menu, select **var://**, and type `local/destination`.

iii. Use the default values in the remaining fields and click **Done**, returning you to the Configure Multi-Protocol Gateway Style Policy view.

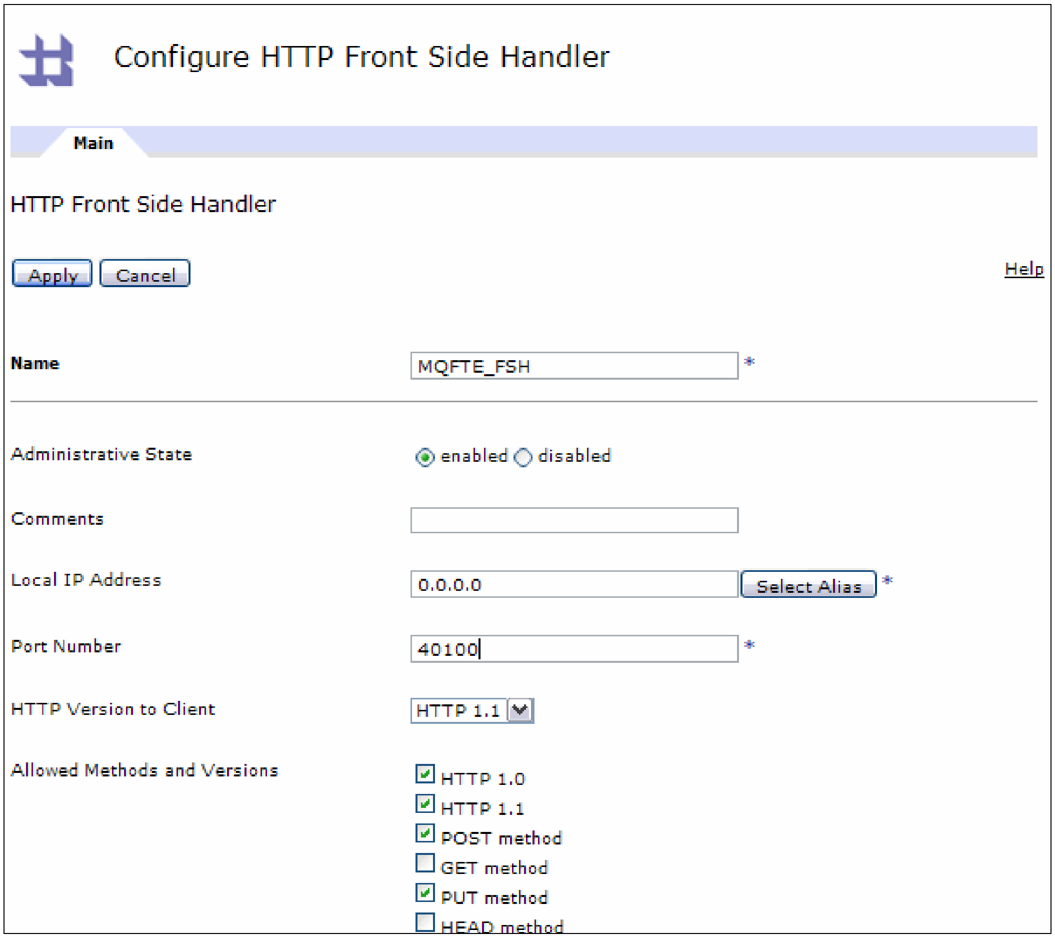
k. Click **Apply Policy** in the Configure Multi-Protocol Gateway Style Policy view.

> **Second Results action:** After clicking Apply Policy, a second Results action is added to the right of the first Results action. This is normal. No configuration of this action is needed.

l. Close the Configure Multi-Protocol Gateway Style Policy view by clicking **Close Window**.

4. You should now be back in the Configure Multi-Protocol Gateway view (Figure 8-13 on page 194). Create an HTTP Front-Side protocol handler for the multi-protocol gateway by clicking the Plus icon in the Front Side Protocol field. Choose **HTTP Front Side** Handler from the list.

   Figure 8-23 illustrates how the configured HTTP front side handler for the multi-protocol gateway should look. Fields that are populated with default values might not display.



Configure HTTP Front Side Handler

HTTP Front Side Handler

Name: MQFTE_FSH *

Administrative State: ● enabled ○ disabled

Comments:

Local IP Address: 0.0.0.0 Select Alias *

Port Number: 40100 *

HTTP Version to Client: HTTP 1.1

Allowed Methods and Versions:
☑ HTTP 1.0
☑ HTTP 1.1
☑ POST method
□ GET method
☑ PUT method
□ HEAD method

Figure 8-23 Configure HTTP Front Side Handler

Complete these fields as follows:

i. In the Configure HTTP Front Side Handler view Main tab type `MQFTE_FSH` in the Name field.

ii. Enter `xb60` or leave it at `0.0.0.0` for the Local IP Address field.

iii. In the Port Number field type an unused port number. For this scenario use `40100`.

iv. Leave the defaults for all other fields and click **Apply**.

m. Click **Apply** in the Configure Multi-Protocol Gateway view to finish the multi-protocol gateway configuration.

n. Save the configuration by clicking **Save Config** in the upper right corner of the view.

8.4.5 Configuring trading partner profiles

The B2B partner profile is the configuration object where the trading partner information is defined. This information includes the profile name, profile type, business IDs, AS security, destinations for document routing, and contact information. A trading relationship consists of, at a minimum, one internal and one external profile. For more detailed information about profile types, see the “B2B Partner Profiles” section in *IBM WebSphere DataPower B2B Appliance XB60 Revealed*, SG24-7745.

Configuring your company’s profile (internal)

The following steps create the internal profile:

1. Click **B2B Partner Profile** from the Control Panel. If you are not in the Control Panel, click it at the top of the Left Navigation menu.

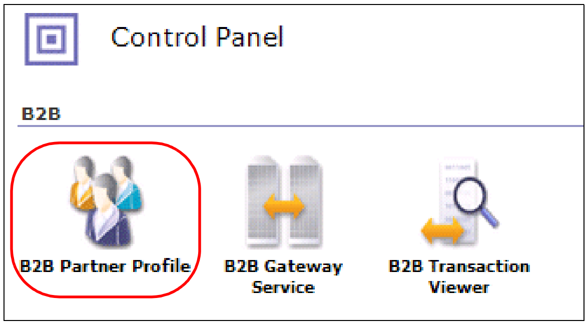
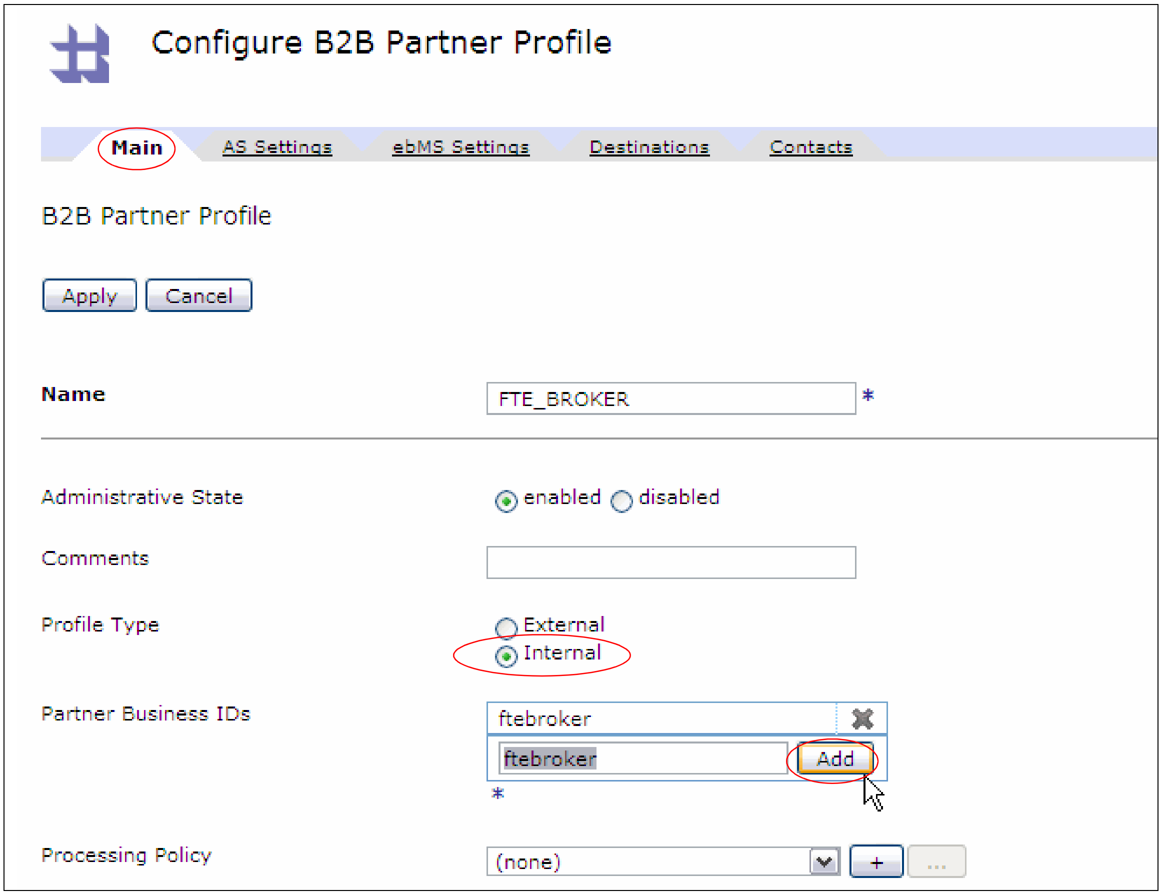


Figure 8-24   B2B portion of the control panel

2. In the Configure B2B Partner Profile list view, click **Add**.

3. Configure the Partner Profile Main tab as shown in Figure 8-25.



Configure B2B Partner Profile

B2B Partner Profile

Name: FTE_BROKER *

Administrative State: ● enabled ○ disabled

Comments:

Profile Type: ○ External ● Internal

Partner Business IDs: ftebroker
ftebroker *

Processing Policy: (none)

Figure 8-25 FTE_BROKER internal profile - Main tab

Complete these fields as follows:

a. In the Name field enter a descriptive name for your internal profile. For this scenario use `FTE_BROKER`.

b. Choose **enabled** in the Administrative State field.

c. Choose **Internal** in the Profile Type field.

d. In the Partner Business IDs field type `ftebroker` and click **Add** to add the business ID to the list.

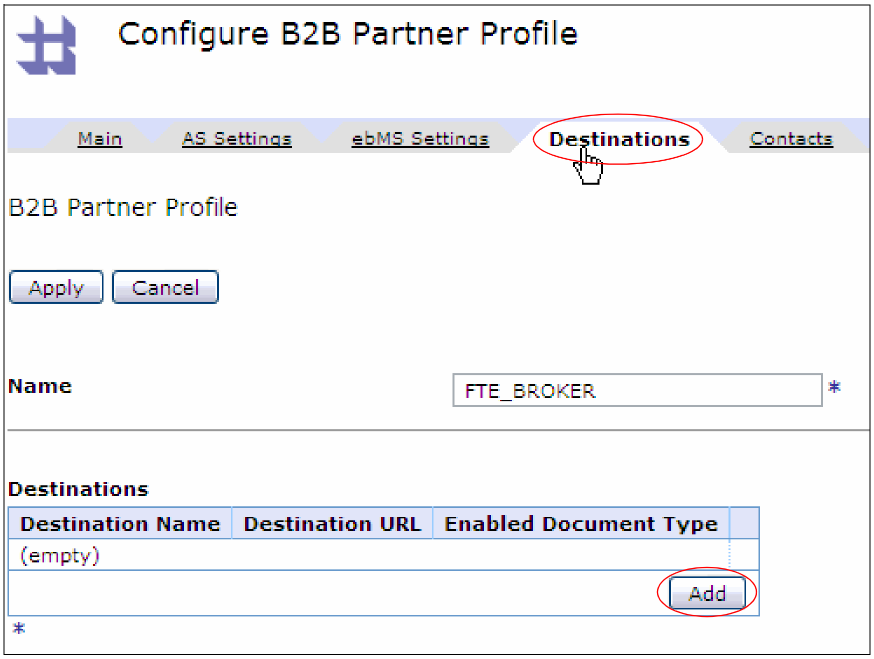
e. Leave the default values in all other fields.

> **Important:** Do not click **Apply** at this time. The other tabs need to be configured first. Proceed to step 4.

4. Configure the Destinations tab.

   a. Click the **Destinations** tab. The Name field carries over to the Destinations view. Do not change it.

   b. In the Destinations view click **Add** to add a destination to this profile (Figure 8-26).



B2B Partner Profile

Name FTE_BROKER *

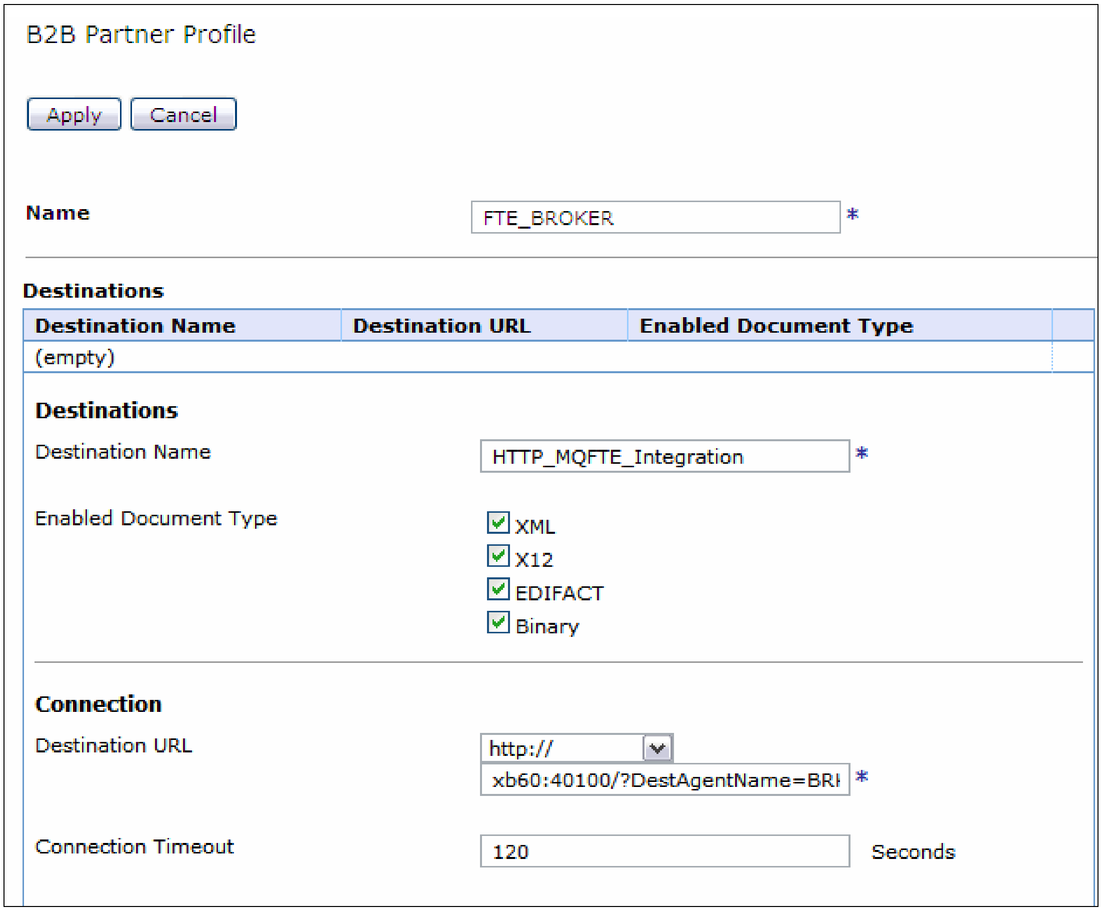
Destinations

Destination Name	Destination URL	Enabled Document Type
(empty)		

*

Figure 8-26 B2BFTE internal profile - Add destination

c. Because this profile is an internal profile, the destination typically is a system or application inside your private network. In this scenario, the destination is the broker via the HTTP front side handler in the multi-protocol gateway (Figure 8-23 on page 202). Complete the fields to define this destination as shown in Figure 8-27.



B2B Partner Profile

Name: FTE_BROKER *

Destinations

Destination Name	Destination URL	Enabled Document Type
(empty)		

Destinations

Destination Name: HTTP_MQFTE_Integration *

Enabled Document Type:
☑ XML
☑ X12
☑ EDIFACT
☑ Binary

Connection

Destination URL: http://
xb60:40100/?DestAgentName=BRI *

Connection Timeout: 120 Seconds

Figure 8-27 Defining a new protocol for FTE_BROKER under Connection in PARTNER domain

Complete these fields as follows:

i. Enter a descriptive name in the Destination Name field. For the purpose of this scenario use `HTTP_MQFTE_Integration`.

ii. Leave all of the boxes checked in the Enable Document Type section. This allows your internal profile to accept and produce all supported file types.

iii. In the Connection section use the drop-down menu to select **http://** as the Destination URL and type and use `xb60:40100/?DestAgentName=BRKSAFE.DEFAULT&DestAgentQMgr=QMBRK` as the address.

> **Troubleshooting tip:** Information to the right of the question mark (?) is used as input to the XSL used in the MQ FTE Proxy multiprotocol gateway. Be sure to type in the URL exactly as seen in this step, as it is case sensitive. If the file is not moved from the `/XB60/ftewmb/In` directory to the destination directory, `C:\FTEin`, the issue is usually related to having incorrect information in this URL.

iv. Change the connection timeout to **120** seconds.

v. Leave the User Name and Password fields blank because we are not using basic authentication.

vi. Click **Apply** inside the Destination configuration view to return to the Destination List.

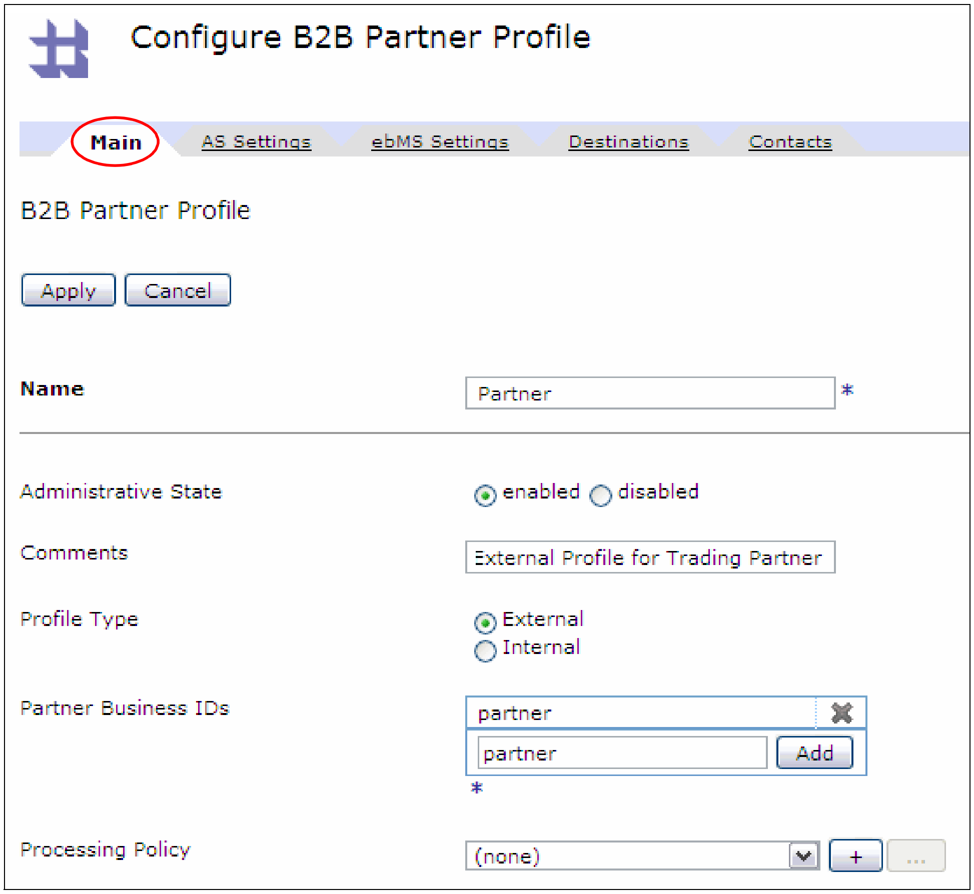
5. Configuring the Partner Profile Contacts tab is optional. We do not configure contacts for this scenario.
6. Now that the internal profile is completely configured, click **Apply** for the profile.
7. Save the profile by clicking **Save Config** in the upper right corner of the web page.

Configuring the trading partner profile (external)

Configuring your external partner's profile is completed in the same manner in which you configured the internal partner profile.

1. After creation of the internal profile you should be back in the B2B Partner Profile list view. If not, expand **Services** in the left navigation menu and click **B2B Partner Profile**.
2. Click **Add** to configure the external profile.
3. Select the **Main** tab.

   Figure 8-28 illustrates how the configured external partner Main tab should look. Fields that are populated with default values might not be displayed.



Name	Partner *
Administrative State	● enabled ○ disabled
Comments	External Profile for Trading Partner
Profile Type	● External ○ Internal
Partner Business IDs	partner partner *
Processing Policy	(none)

Figure 8-28 External partner profile - Main tab

Complete these fields as follows:

a. In the Name field enter a descriptive name for your partner's external profile. For this scenario use `Partner`.

b. Choose **enabled** in the Administrative State field.

c. Choose **External** in the Profile Type field.

d. In the Partner Business IDs field enter your ID. For this scenario use `partner` and click **Add** to add it to the list. Leave the defaults in all remaining fields.

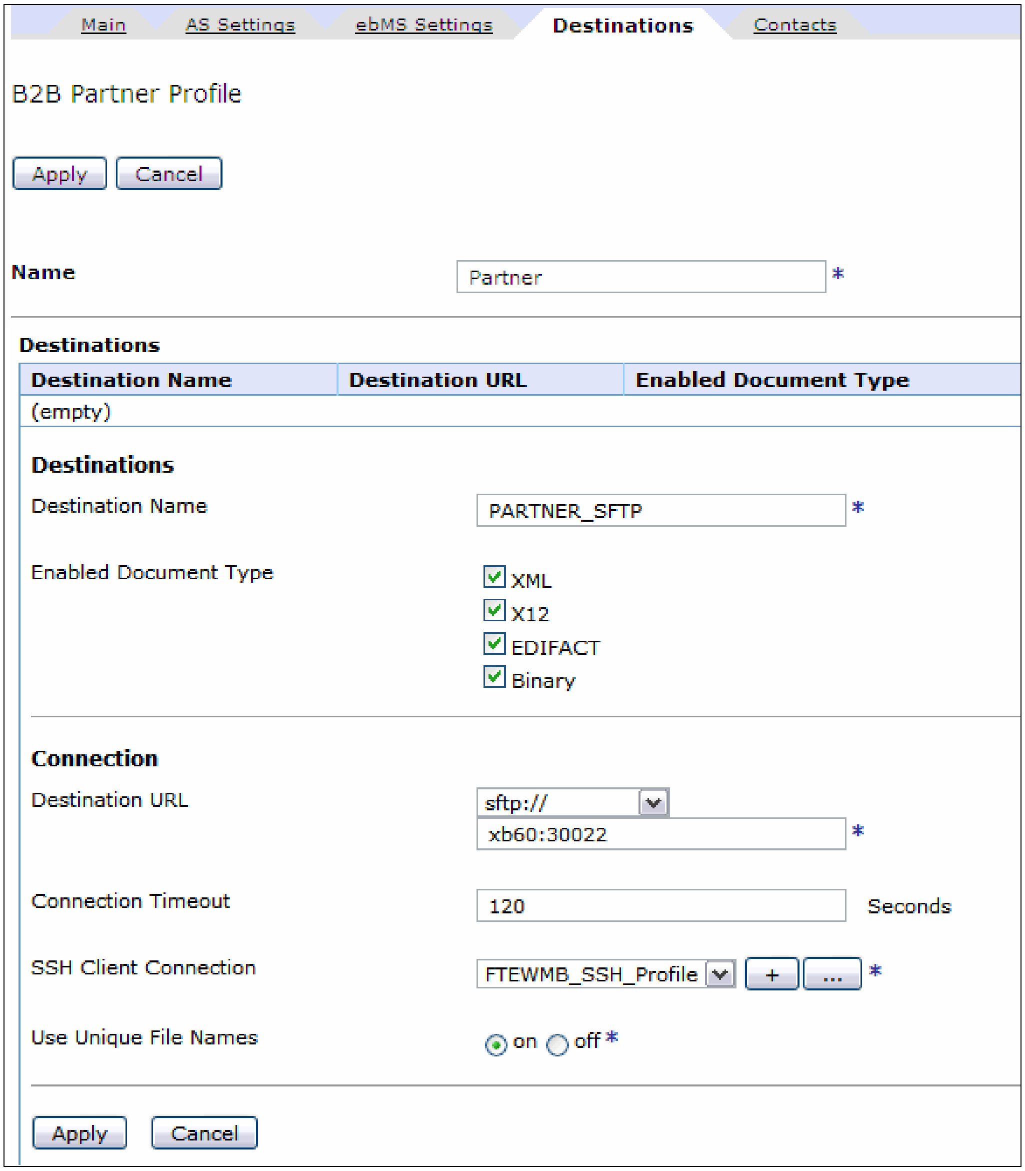
> **Important:** Do not click **Apply** at this time. The other tabs need to be configured first. Proceed to step 4.

4. Configure the destination.

   The partner profile must be configured for connections to the SFTP server defined in the PARTNERHUB domain (the external partner).

   a. Click the **Destinations** tab, then click **Add** to create a new destination.

b. Configure the destination as shown in Figure 8-29.



B2B Partner Profile

Name: Partner *

Destinations

Destination Name	Destination URL	Enabled Document Type
(empty)		

Destinations

Destination Name: PARTNER_SFTP *

Enabled Document Type: ☑ XML ☑ X12 ☑ EDIFACT ☑ Binary

Connection

Destination URL: sftp:// xb60:30022 *

Connection Timeout: 120 Seconds

SSH Client Connection: FTEWMB_SSH_Profile *

Use Unique File Names: ● on ○ off *

Figure 8-29 Defining a new protocol for PARTNER under Connection

c. In the Destination Name field, enter `PARTNER_SFTP`.

d. Select **sftp://** in the pull-down menu next to Destination URL in the Connection section.

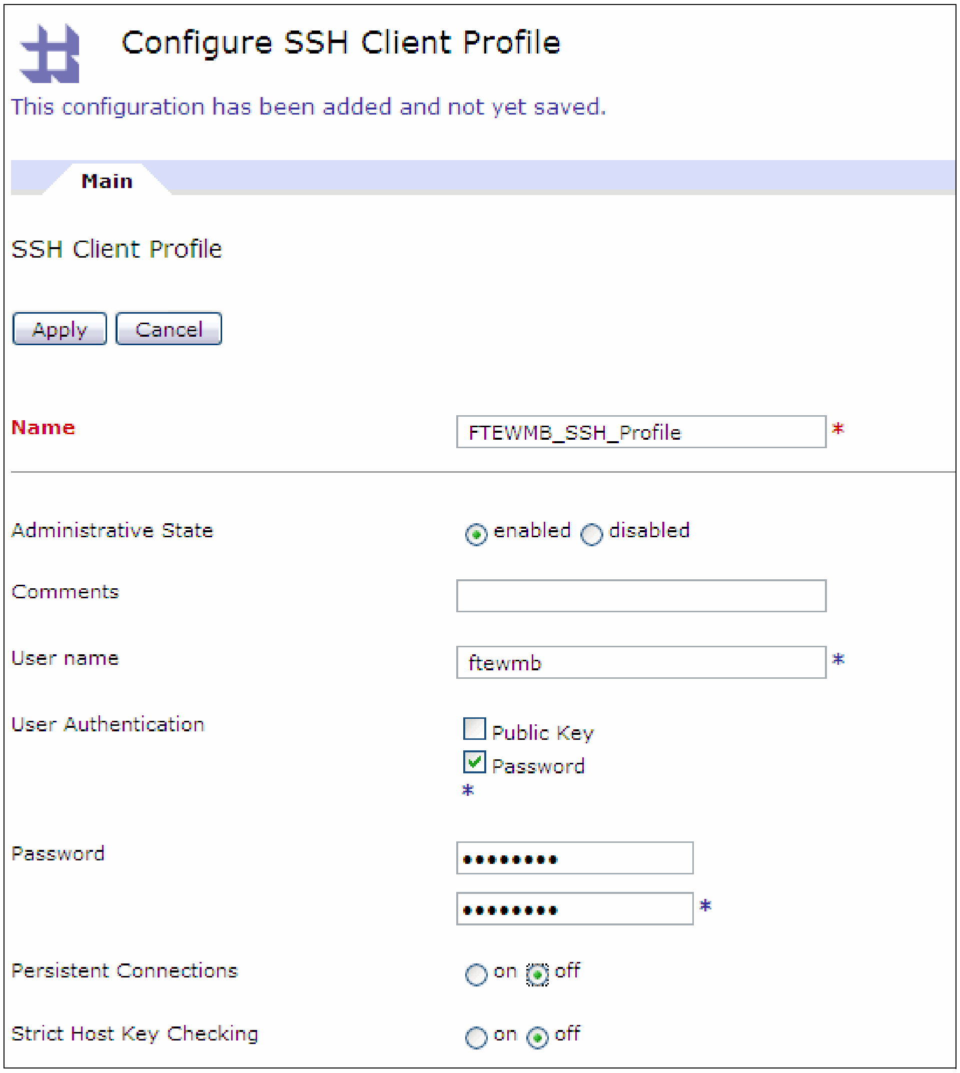
e. Type `xb60:30022` in the box under Destination URL to specify the address (host name and port) of the SFTP server.

f. Change the Connection Timeout field to 120.

g. Add an SSH Client Connection profile:

i. Click the Plus icon to the right of the SSH Client Connection input field.

ii. Enter the values shown in Figure 8-30. The user name and password are used to access the partner's SFTP server, B2B gateway, or both. For this scenario we use user ID ftewmb and password itso4you. Click **Apply**.



Configure SSH Client Profile

This configuration has been added and not yet saved.

SSH Client Profile

Name	FTEWMB_SSH_Profile *
Administrative State	● enabled ○ disabled
Comments	
User name	ftewmb *
User Authentication	□ Public Key ☑ Password *
Password	•••••••• •••••••• *
Persistent Connections	○ on ● off
Strict Host Key Checking	○ on ● off

Figure 8-30 SSH client profile configuration

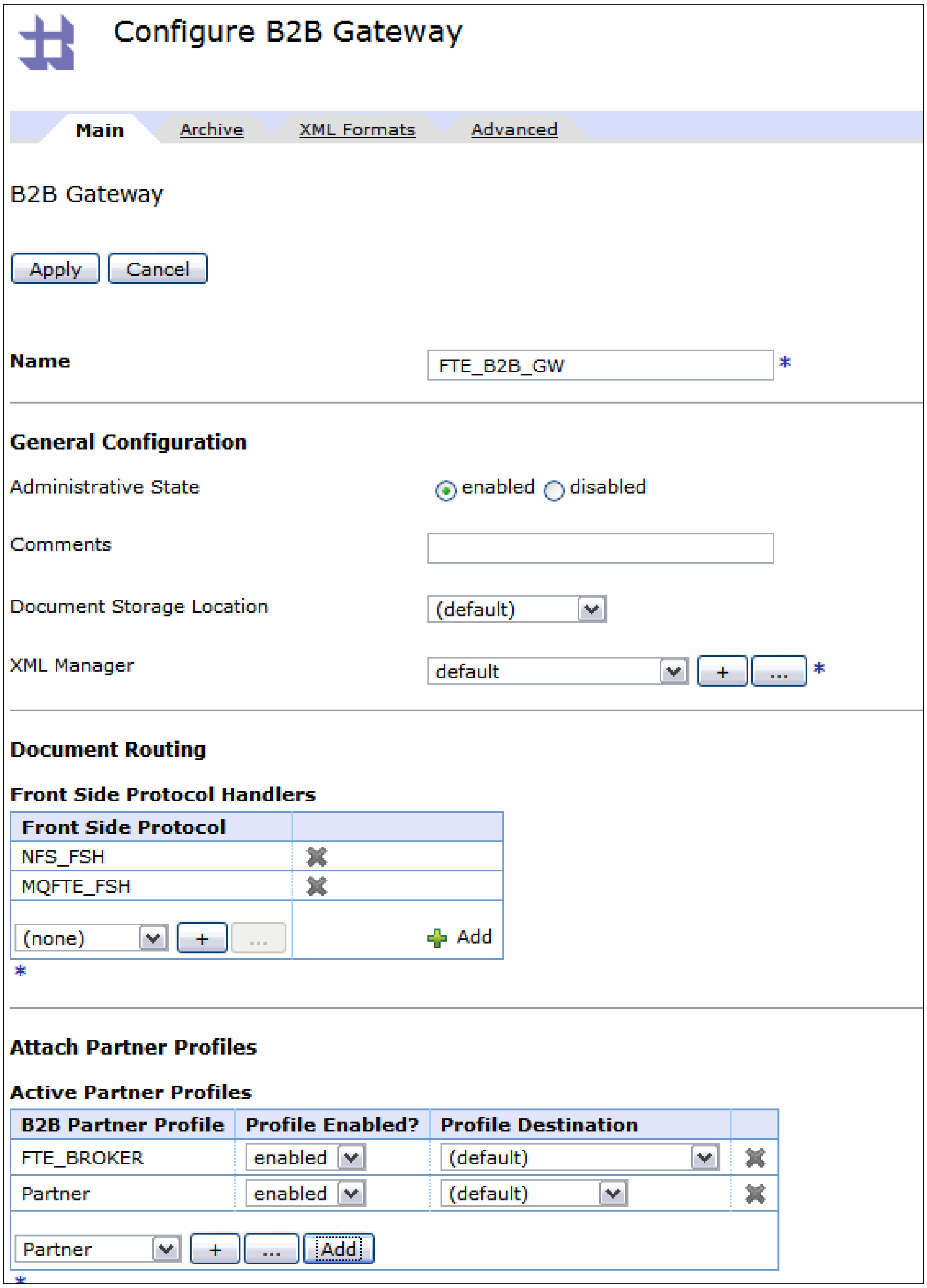
h. Back on the Destinations tab, leave the defaults in the remaining fields and click **Apply**.

5. Configuring the Partner Profile Contacts tab is optional. We do not configure contacts for this scenario.
6. Now that the profile is completely configured click **Apply** for the profile.
7. Save the profile by clicking **Save Config** in the upper right corner of the web page.

8.4.6 Configuring the B2B gateway

Your B2B gateway is the primary business-to-business hub and is depicted in the scenarios as the owner of the XB60. In this section, we configure the FTE_B2B_GW B2B gateway to trade with a single partner's business-to-business hub. The partner's business-to-business hub can be any SFTP-interoperable product. However, for the purpose of this exercise we simulated the partner's business-to-business hub in a separate domain on the same XB60 being used for this scenario.

1. Expand **Services** in the left navigation menu and click **B2B Gateway Service**.
2. In the Configure B2B Gateway list view, click **Add**.
3. Configure the B2B Gateway Main tab. Figure 8-29 on page 209 shows the final configuration. The next steps help you complete this configuration.



Name: FTE_B2B_GW *

General Configuration

Administrative State: ● enabled ○ disabled

Comments:

Document Storage Location: (default)

XML Manager: default *

Document Routing

Front Side Protocol Handlers

Front Side Protocol	
NFS_FSH	✖
MQFTE_FSH	✖
(none)	Add

*

Attach Partner Profiles

Active Partner Profiles

B2B Partner Profile	Profile Enabled?	Profile Destination	
FTE_BROKER	enabled	(default)	✖
Partner	enabled	(default)	✖
Partner Add			

*

Figure 8-31 FTE_B2B_GW gateway - Main tab

Complete these fields as follows:

a. Enter the B2B gateway name in the Name field. For this scenario use `FTE_B2B_GW`.

b. Choose **enabled** in the Administrative State field.

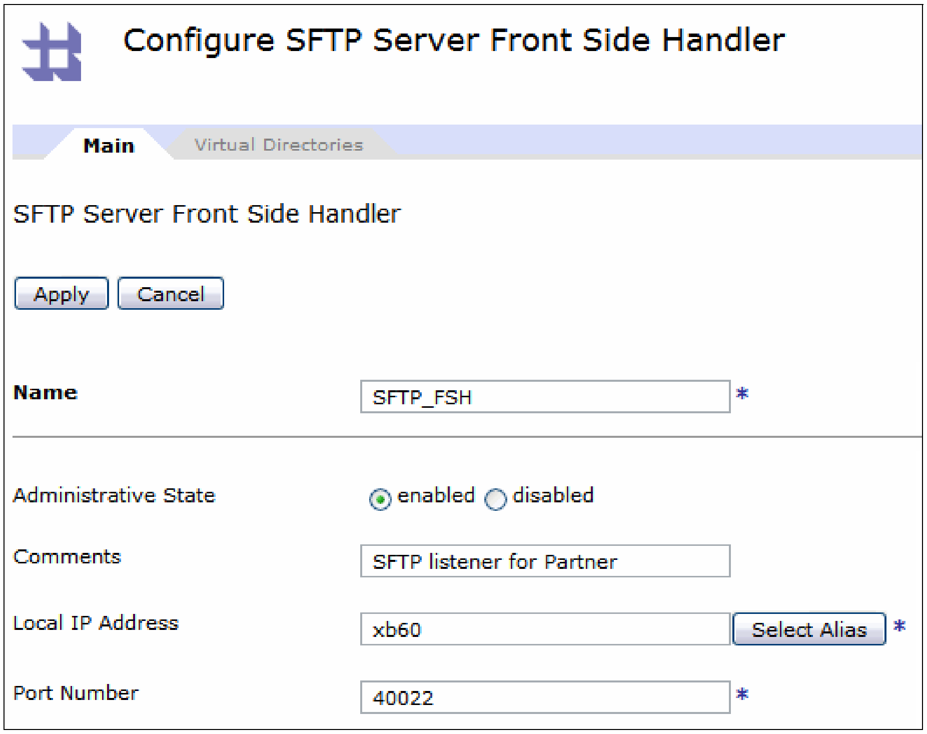
c. Optional: Add comments that describe this gateway.

d. Accept the defaults for the Document Storage Location and XML Manager fields.

e. In the Document Routing section of the Main tab create and configure an SFTP protocol handler (to be used as a SFTP listener for inbound files from the external partner).

   i. Click the Plus icon in the Front Side Protocol field (Figure 8-31 on page 211).

   ii. Choose **SFTP Server Front Side Handler** from the list. This opens the Configure SFTP Server Front Side Handler view (Figure 8-32). Select the **Main** tab.



Configure SFTP Server Front Side Handler

SFTP Server Front Side Handler

Name SFTP_FSH *

Administrative State ● enabled ○ disabled

Comments SFTP listener for Partner

Local IP Address xb60 *

Port Number 40022 *

Figure 8-32 B2BFTE_HUB gateway - Main tab

   iii. Enter `SFTP_FSH` in the Name field.

   iv. Enter `SFTP listener for Partner` in the Comments field.

   v. In the Local IP Address field type `xb60`.

   vi. In the Port Number field type an unused port number. For this scenario use `40022`.

   vii. Leave the defaults for all other fields and click **Apply**.

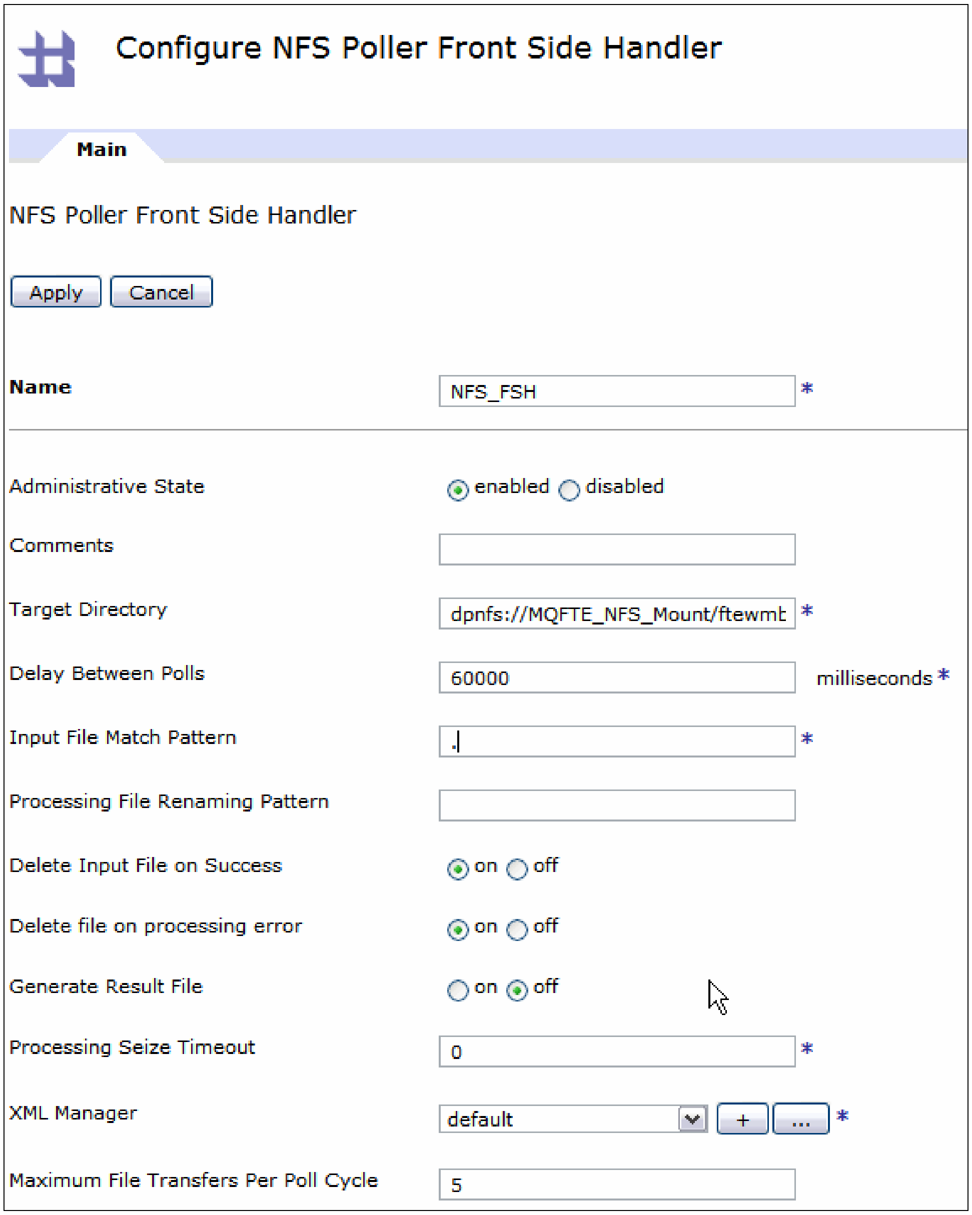
   viii.Click the **+ Add** link to add the poller to the Front Side Protocol list.

f. In the Document Routing section of the Main tab create and configure an NFS Poller Front Side Handler. This handler is used to receive files from the NFS file share that is associated with WebSphere MQ File Transfer Edition.

   i. Click the Plus icon to create a new handler.

   ii. Choose **NFS Poller Front Side Handler** from the list of handlers.

This opens the Configure NFS Poller Front Side Handler view. Figure 8-33 illustrates how the configured NFS front side handler should look. Fields that are populated with default values might not display.


Configure NFS Poller Front Side Handler
Main
NFS Poller Front Side Handler
Apply
Cancel

Name	NFS_FSH *
Administrative State	● enabled ○ disabled
Comments	
Target Directory	dpnfs://MQFTE_NFS_Mount/ftewmb *
Delay Between Polls	60000 milliseconds *
Input File Match Pattern	. *
Processing File Renaming Pattern	
Delete Input File on Success	● on ○ off
Delete file on processing error	● on ○ off
Generate Result File	○ on ● off
Processing Seize Timeout	0 *
XML Manager	default + ... *
Maximum File Transfers Per Poll Cycle	5

Figure 8-33 FTE_B2B_GW gateway - NFS Poller front side protocol handler

iii. In the Main tab, in the Name field enter `NFS_FSH`.

iv. Choose **enabled** in the Administrative State field.

v. Optional: Add comments that describe this handler.

vi. In the Target Directory field enter `dpnfs://MQFTE_NFS_Mount/ftewmb/Out/`.

vii. In the Delay Between Polls field, select the amount of time to delay between polling intervals. The default is 60000 milliseconds.

viii.In the Input File Match Pattern type a period (`.`) to match all.

ix. Make sure that the Delete Input File on Success is on. Files must be deleted after they are picked up, otherwise they are picked again during the next poll cycle.

x. Make sure that the Delete File on Processing Error is on. Files must be deleted after they are picked up, otherwise they are picked again during the next poll cycle. It is common for the file to error because the partner's hub is down. We want these failed transfers to be resent to the partner from the gateway, not kept in the directory.

xi. Make sure that the Generate Results File is off.

xii. Accept the default values for all other fields and click **Apply**.

xiii.Click the **+ Add** link to add the poller to the Front Side Protocol list.

> **Troubleshooting tip:** The NFS Poller Front Side Handler polls the NFS directory that is shared with WebSphere MQ File Transfer Edition. The poller looks for files to send to the external partner's SFTP server. If your NFS mount point is inside the protected network, be sure that the inner firewall has a rule that allows the XB60 to send data over the NFS ports being used (typically, 2049 and 111). Your internal network security policies govern where the NFS share is located and how it is protected. If files are not being picked up, your target directory value might be wrong or BRKSAFE.DEFAULT might be down and is not sending files to `/XB60/ftewmb/Out`.

g. Back on the Main tab of the Configure B2B Gateway view (Figure 8-31 on page 211), find the Attach Partner Profiles section.

i. Click the drop-down menu, select **FTE_BROKER**, and click **Add**.
ii. Click the drop-down menu again, select **Partner**, and click **Add**.

This associates the profiles that you created in 8.4.5, "Configuring trading partner profiles" on page 203, with your B2B gateway.

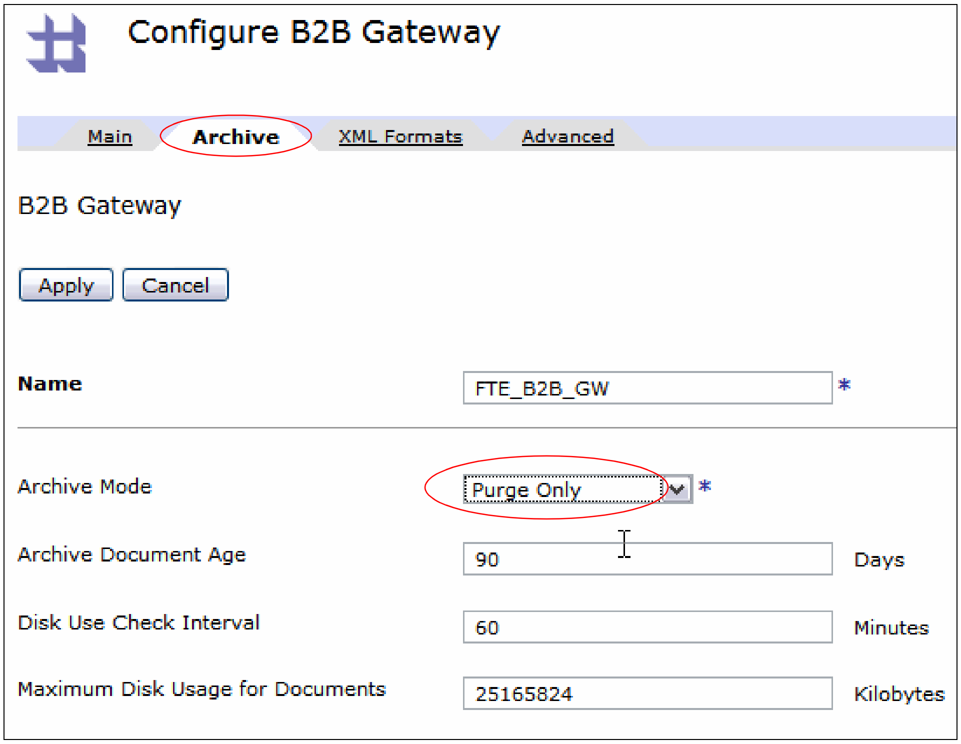
h. Skip active profile groups. We do not use them for this scenario.

> **Important:** Do not click **Apply** at this time. The other tabs must be configured first.

4. Configure the **Archive tab** for the B2B gateway.

> **Troubleshooting tip:** The Archive tab is used to automatically keep the B2B document and metadata storage areas clean. There are two modes:
>
> - Archive and purge
> - Purge only
>
> Perform capacity planning to determine how much drive space you need to support your retention policies. In certain cases you will need more space than is available on the DataPower device's hard drive, and you will need to store business-to-business payloads off device using one of the external hard drive storage options. See the XB60 user documentation for details on using external drives with the appliance. The user documentation for the XB60 can be found at the following URL:
>
> `http://publib.boulder.ibm.com/infocenter/wsdatap/v3r8m1/index.jsp?topic=/xb60/welcome.htm`

Figure 8-34 illustrates how the configured B2B Gateway Archive tab should look. Fields that are populated with default values might not display.



B2B Gateway

Name: FTE_B2B_GW *

Archive Mode: Purge Only *

Archive Document Age: 90 Days

Disk Use Check Interval: 60 Minutes

Maximum Disk Usage for Documents: 25165824 Kilobytes

Figure 8-34 FTE_B2B_GW gateway - Archive tab

Complete these fields as follows:

a. The Name field carries over to the Archive panel. Do not change it.

b. In the Archive Mode field use the drop-down menu and select **Purge Only**.

c. Accept the defaults for all of the other fields.

5. Click the **XML Formats** tab. This tab is used to configure the XPath's of the sender and receiver ID for XML documents that are to be processed through this gateway.

> **Additional materials:** The `StkReplReq.xml` file is included in Appendix C, "Additional material" on page 317, in the `BrokerScenario_Files` directory.

   a. The Name field carries over to the XML Formats panel. Do not change it.

   b. In the XPath Routing Policies (Figure 8-35) field click the Plus icon to add a new XPath Routing Policy.

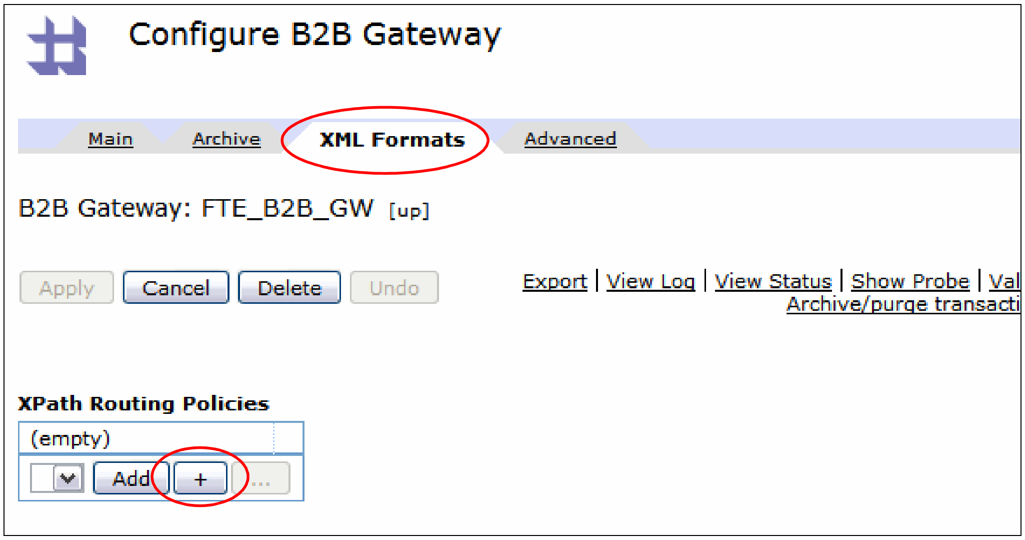
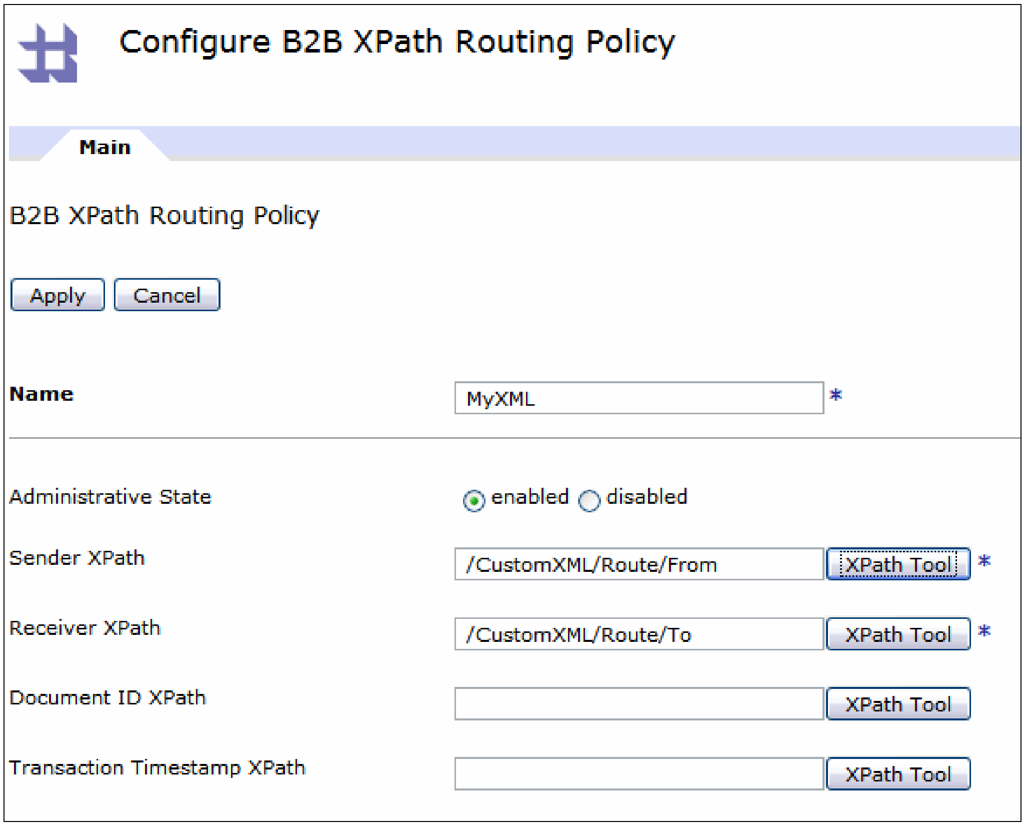


Figure 8-35 Configure B2B Gateway XML Formats tab

The Configure B2B XPath Routing Policy panel opens. Figure 8-36 shows the final configuration of the policy. The next steps describe how to complete this configuration.



B2B XPath Routing Policy

Name: MyXML *

Administrative State: ● enabled ○ disabled

Sender XPath: /CustomXML/Route/From *

Receiver XPath: /CustomXML/Route/To *

Document ID XPath:

Transaction Timestamp XPath:

Figure 8-36 Configure B2B XPath Routing Policy

Complete these fields as follows:

i. In the Name field enter a descriptive name, in this case, `MyXML`.

ii. Choose **enabled** in the Administrative State field.

iii. In the Sender XPath field click **XPath Tool** to upload the XML file and extract the XPath that we need. Example 8-1 shows an extract of the contents.

Example 8-1 StkReplReq.xml

```
<?xml version="1.0" ?>
 <CustomXML>
   <Route>
    <To>ftebroker</To>
    <From>partner</From>
   </Route>
   <StockReplenishment>
    <BranchNumber>123</BranchNumber>
    <SequenceNumber />
   <LineItem>
     <ItemID>AA0534</ItemID>
     <Description>2oz White Chocolate</Description>
     <MerchandiseHierarchy Level="Department">Chocolates</MerchandiseHierarchy>
     <UnitListPrice ForeignAmount="0.87" Currency="GBP">1.64</UnitListPrice>
     <RegularSalesUnitPrice>1.31</RegularSalesUnitPrice>
     <ActualSalesUnitPrice>1.31</ActualSalesUnitPrice>
     <ExtendedAmount>3.62</ExtendedAmount>
     <Quantity>63</Quantity>
   </LineItem>
 </StockReplenishment>
</CustomXML>
```

iv. In the Build XPath Expression from sample XML file panel choose **Upload**.

v. In the Upload File panel be sure that the source is File. Click **Browse** in the File to Upload field, navigate to your XML file, and click **Open**.

vi. Click **Upload** on the bottom left of the Upload File panel and click **Continue** in the upload success box. This puts you back on the Build XPath Expression from sample XML file panel.

vii. The contents of the XML file display in the window at the bottom of the panel. If you do not see the XML file displayed, make sure that the file name is selected in the URL of Sample Document field.

viii.Click **<From> Element** and you will see the XPath in the Select XPath Expression box. Then click **Done** to accept the XPath expression. This puts you back on the Configure B2B XPath Routing Policy panel.

ix. For the Receiver XPath, you can use the XPath Tool or simply copy the Sender XPath, paste it into the Receiver XPath field, and change the word `From` to the word `To`.

x. Leave the remaining XPath fields blank and click **Apply** to save the XPath policy. This returns you to the XPath Formats tab, where you will see that the new format is the first item in the list.

6. Now that the B2B gateway is completely configured, save the service by clicking **Apply**.
7. Click **Save Config** to persist your changes.

8.5 Configuring WebSphere Message Broker

> **Additional material:** Appendix C, “Additional material” on page 317, contains a compressed file that contains the message flows for this scenario. The project can be imported into WebSphere Message Broker Toolkit. The compressed file can be found in the `BrokerScenario_Files` folder.

This section shows how to configure WebSphere Message Broker for integration with WebSphere MQ File Transfer Edition and the XB60. Installation of WebSphere Message Broker is not included here. For installation of WebSphere Message Broker see the WebSphere Message Broker V7 information center at:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v7r0m0/index.jsp

This section assumes that the reader has knowledge of building and deploying message flows in WebSphere Message Broker Toolkit. The following topics are covered in this configuration:

- 8.5.1, “Defining the broker” on page 219
- 8.5.2, “Integrating the WebSphere MQ File Transfer Edition with the WebSphere Message Broker” on page 222
- 8.5.3, “Creating message flows with FTE nodes” on page 224

8.5.1 Defining the broker

The broker used in this scenario was defined with the following steps:

1. In WebSphere MQ Explorer, right-click **Brokers** and select **New** → **Local Broker** (Figure 8-37).

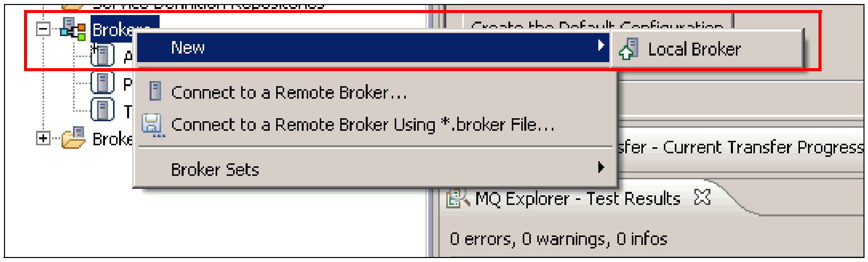
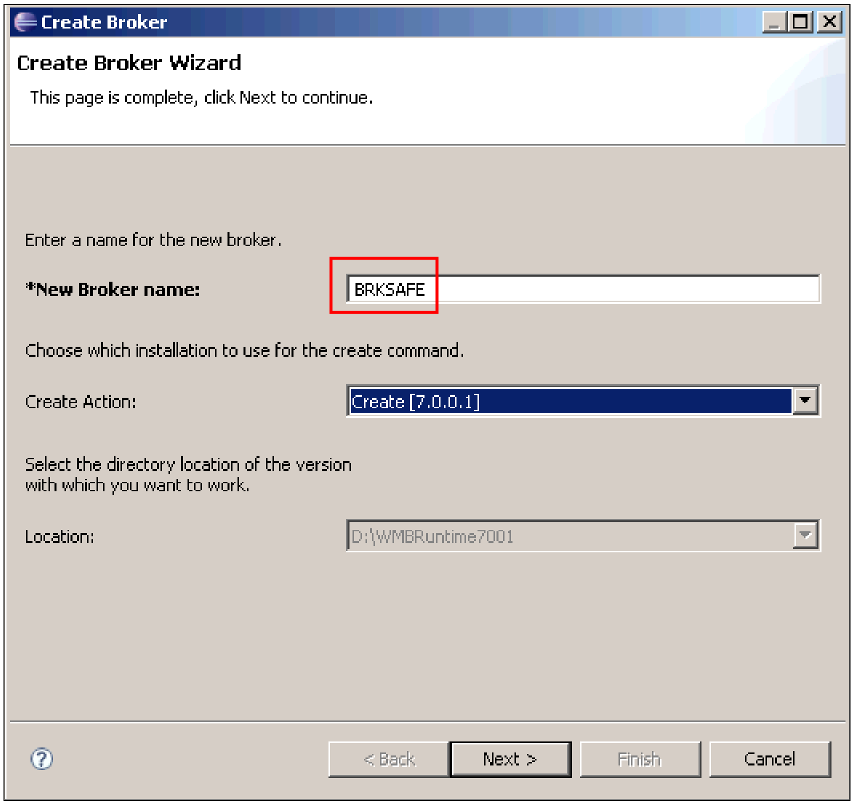

Brokers
New
Local Broker
Connect to a Remote Broker...
Connect to a Remote Broker Using *.broker File...
Broker Sets
sfer - Current Transfer Progress
MQ Explorer - Test Results
0 errors, 0 warnings, 0 infos

Figure 8-37 Creating a new broker using WebSphere Message Broker Explorer

2. Figure 8-38 shows the Create Broker Wizard. In the New Broker name field enter the name for the new broker. In our case we use the name BRKSAFE to indicate that the broker is running in the protected (safe) zone behind the internal firewall. Figure 8-2 on page 182 shows the overall view of the inbound flow.



Create Broker

Create Broker Wizard

This page is complete, click Next to continue.

Enter a name for the new broker.

***New Broker name:** BRKSAFE

Choose which installation to use for the create command.

Create Action: Create [7.0.0.1]

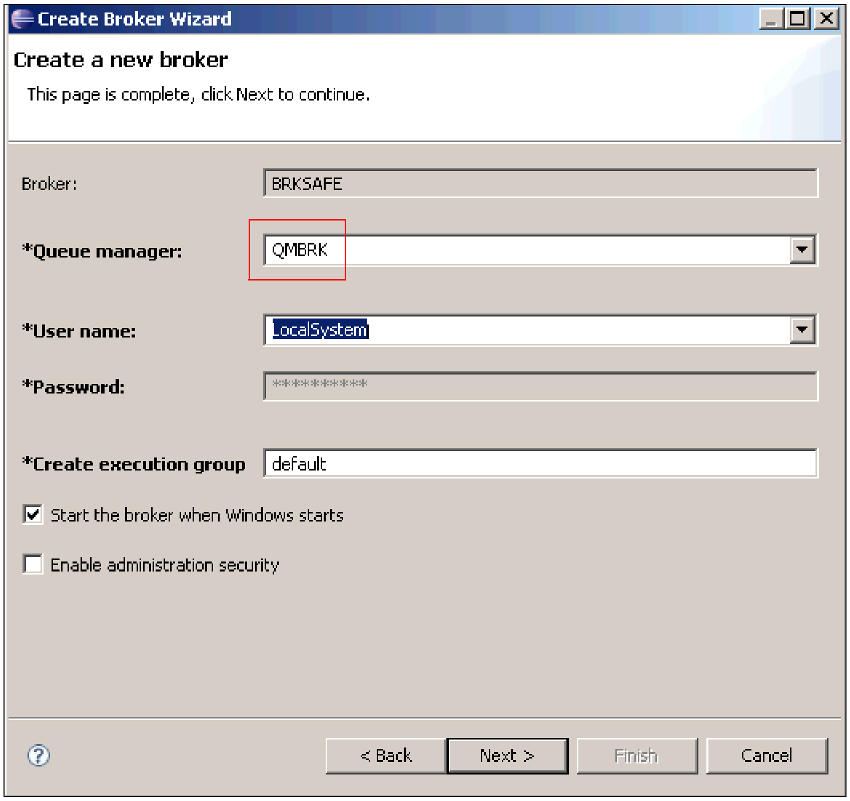
Select the directory location of the version with which you want to work.

Location: D:\WMBRuntime7001

Figure 8-38 Create Broker wizard

Click **Next**.

3. In in the panel shown in Figure 8-39, specify the queue manager where the broker's queues will be defined, QMBRK. Leave the user name and password set to the defaults.



Create a new broker

This page is complete, click Next to continue.

Broker: BRKSAFE

*Queue manager: QMBRK

*User name: LocalSystem

*Password: ***********

*Create execution group default

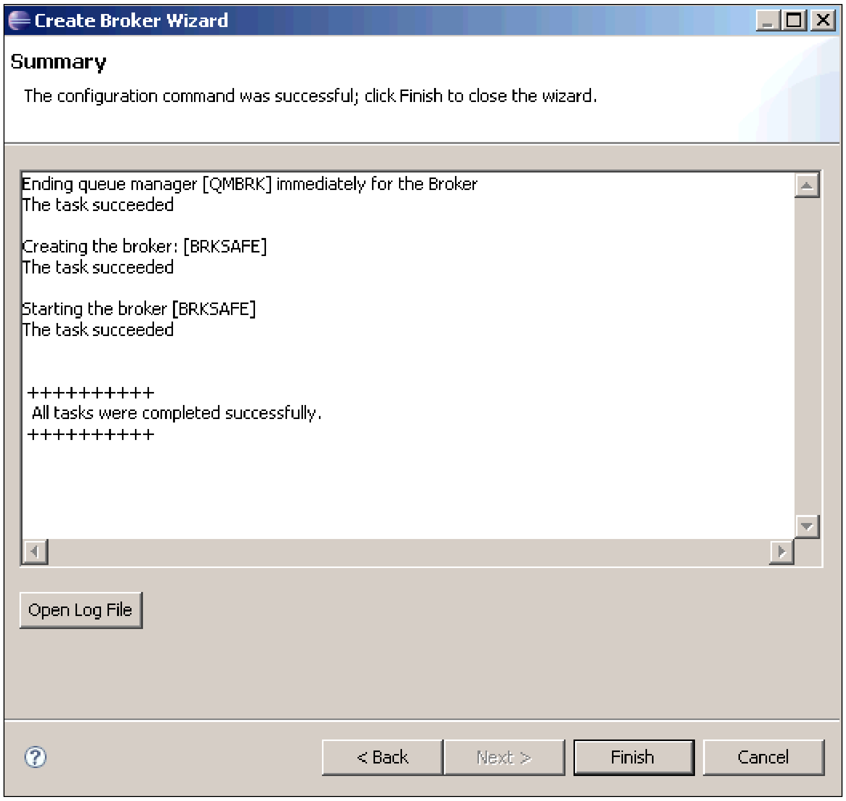
☑ Start the broker when Windows starts

☐ Enable administration security

Figure 8-39 Completed Create Broker Wizard

Click **Next** and you will see the progress bar move as the broker is defined.

4. When the broker creation is complete, you will see a successful completion message (Figure 8-40).



Summary

The configuration command was successful; click Finish to close the wizard.

```
Ending queue manager [QMBRK] immediately for the Broker
The task succeeded

Creating the broker: [BRKSAFE]
The task succeeded

Starting the broker [BRKSAFE]
The task succeeded


 ++++++++++
  All tasks were completed successfully.
 ++++++++++
```

Figure 8-40 Summary - Success

5. You can now see your new broker in WebSphere MQ Explorer in the list of Brokers (Figure 8-41). You can also see that the default execution group for the new broker started.

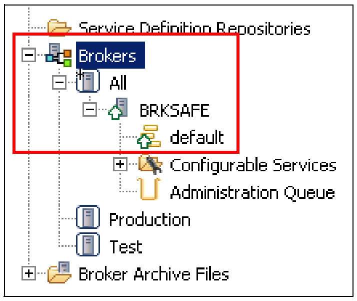


Figure 8-41 WebSphere MQ Explorer with WebSphere Message Broker Explorer plug-in

8.5.2 Integrating the WebSphere MQ File Transfer Edition with the WebSphere Message Broker

The FTEInput and FTEOutput nodes introduced in WebSphere Message Broker V7.0.0.1 provide seamless integration with your existing MQ FTE backbone network. Figure 8-42 shows a typical MQ FTE backbone network with a WebSphere Message Broker deployed with FTE nodes.

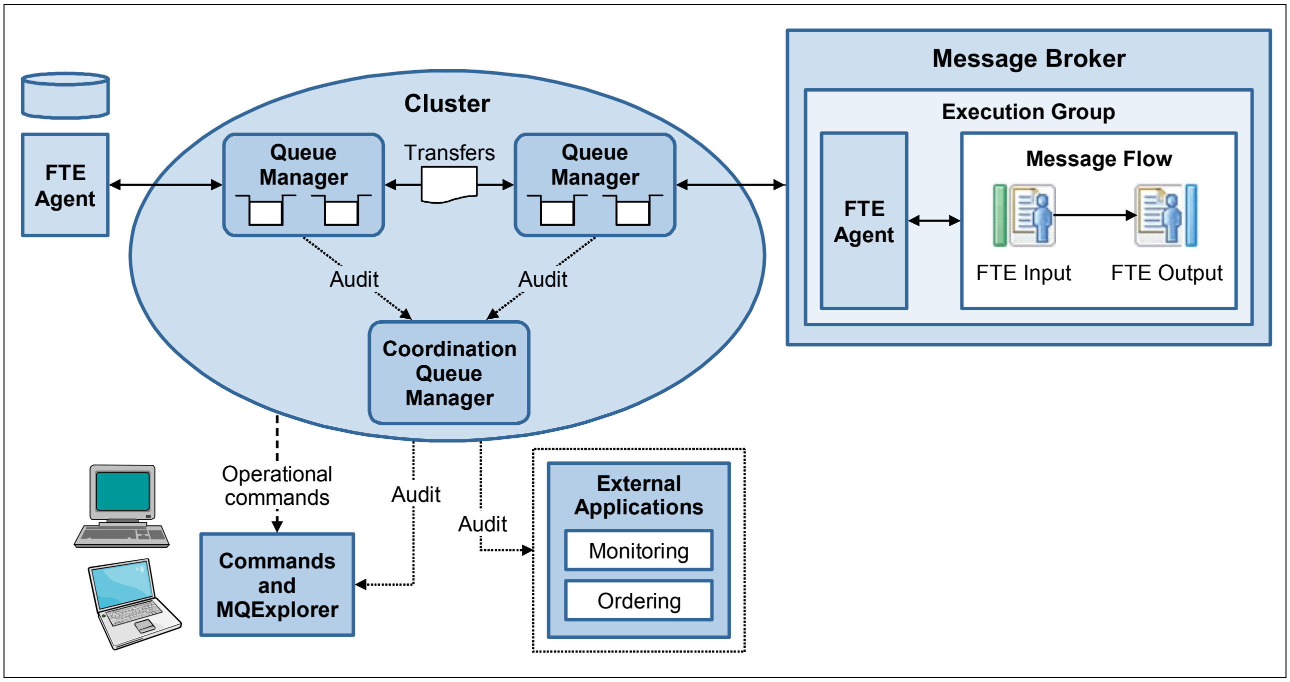


Figure 8-42 Integrating WebSphere MQ File Transfer Edition with WebSphere Message Broker overview

As depicted in Figure 8-42, a WebSphere MQ File Transfer Edition agent runs in each execution group that has deployed flows containing FTE nodes. The agent is responsible for receiving and initiating all WebSphere MQ File Transfer Edition transfers.

You do not need to start or stop this agent. If a flow containing FTE nodes is deployed, the agent is running. The agent is stopped only when the execution group is stopped. The broker queue manager is used as the queue manager for the agent.

Agent name

To send a file to a given execution group, users need to know the name of the agent that the broker creates. The agent name (BRKSAFE.DEFAULT in our case) is derived from *Broker.ExecutionGroup*, and is not configurable. The total name length is limited to 28 characters, with a maximum of 12 characters for the broker name and 15 characters for the execution group. Broker and execution group names longer than these limits are truncated to form the agent name. The name must be in a valid format for generating the MQ Series queue name. Ensure that:

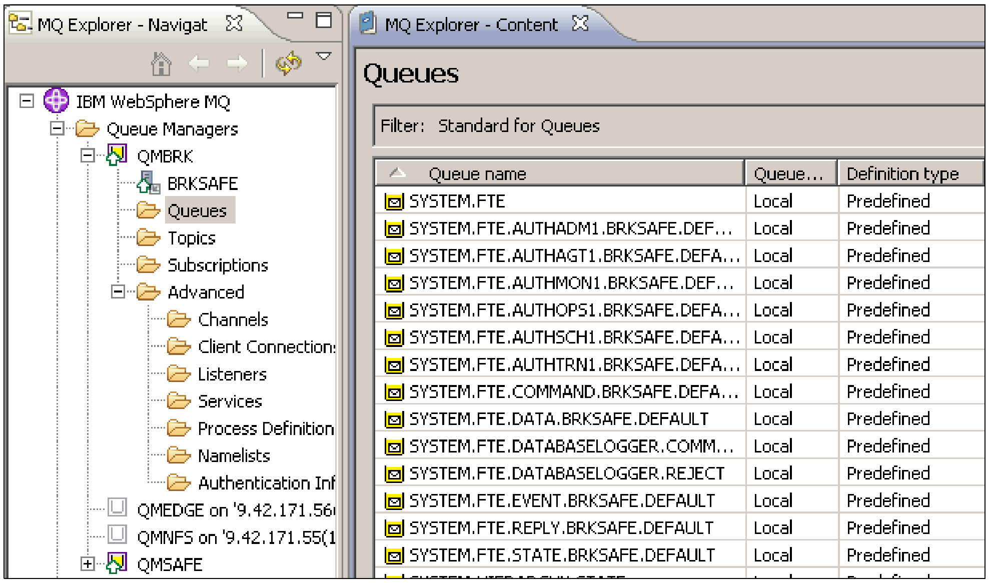
- The broker name is 12 characters or fewer (or at least unique in the first 12 characters).
- The execution group names are 15 characters or fewer (or at least unique in the first 15 characters).

- The broker and execution groups do not contain any characters that are invalid for queue names.
- The broker.executiongroup tuples are all unique, even if the case is ignored.

Queue manager FTE artifacts

When a message flow with FTE nodes is deployed, the broker creates all the required artifacts on the queue manager for the agent (Figure 8-43) and on the broker queue manager when it is the coordination queue manager. If the artifact creation fails due to the configuration of the system or permissions, the broker might not be able to create all artifacts. In this case, the user must create them in advance manually or using scripts. In our scenario, the BRKSAFE broker created all the queues necessary for the BRKSAFE.DEFAULT agent. The coordination queue manager was created previously and needed no additional artifacts.

> **Database logger queues:** When the broker creates the MQ artifacts, it also defines the queues for the database logger, because by default, the broker assumes that its queue manager is the coordination queue manager.

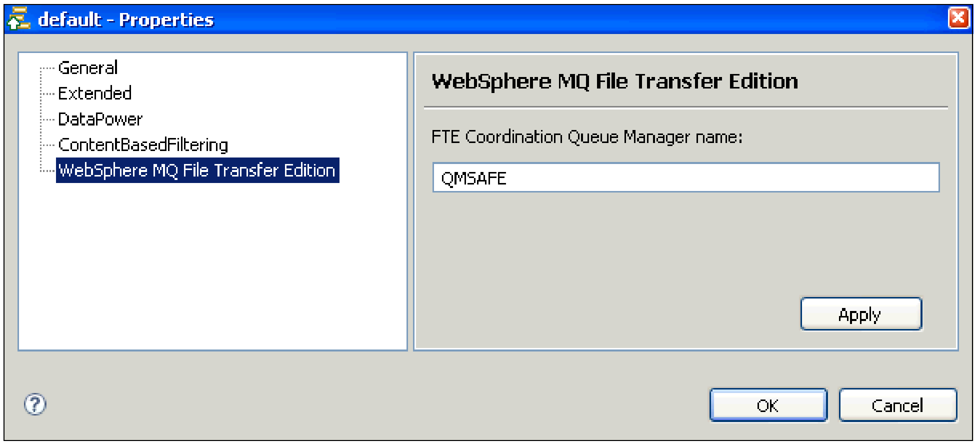


Queue name	Queue...	Definition type
SYSTEM.FTE	Local	Predefined
SYSTEM.FTE.AUTHADM1.BRKSAFE.DEF...	Local	Predefined
SYSTEM.FTE.AUTHAGT1.BRKSAFE.DEFA...	Local	Predefined
SYSTEM.FTE.AUTHMON1.BRKSAFE.DEF...	Local	Predefined
SYSTEM.FTE.AUTHOPS1.BRKSAFE.DEFA...	Local	Predefined
SYSTEM.FTE.AUTHSCH1.BRKSAFE.DEFA...	Local	Predefined
SYSTEM.FTE.AUTHTRN1.BRKSAFE.DEFA...	Local	Predefined
SYSTEM.FTE.COMMAND.BRKSAFE.DEFA...	Local	Predefined
SYSTEM.FTE.DATA.BRKSAFE.DEFAULT	Local	Predefined
SYSTEM.FTE.DATABASELOGGER.COMM...	Local	Predefined
SYSTEM.FTE.DATABASELOGGER.REJECT	Local	Predefined
SYSTEM.FTE.EVENT.BRKSAFE.DEFAULT	Local	Predefined
SYSTEM.FTE.REPLY.BRKSAFE.DEFAULT	Local	Predefined
SYSTEM.FTE.STATE.BRKSAFE.DEFAULT	Local	Predefined

Figure 8-43 WebSphere MQ queues for broker agent BRKSAFE.DEFAULT

Specifying a coordination queue manager

By default, the broker queue manager is the coordination queue manager. You can specify a different coordination queue manager by using the WebSphere Message Broker Explorer or the **mqsichangeproperties** command. In our scenarios, we define the coordination queue manager as QMSAFE instead of the broker's queue manager, QMBRK. Figure 8-44 shows the panel for changing the coordination queue manager property. To find this panel in WebSphere Message Broker Explorer, right-click the execution group and select **Properties** → **WebSphere MQ File Transfer Edition**.



FTE Coordination Queue Manager name: QMSAFE

Figure 8-44 Execution group default properties panel - Setting coordination queue manager

Directory for file transfers

WebSphere Message Broker uses a location in its work path to store transfers to remote agents. It uses another location as the default directory for received files. The high-level directory path for both locations is:

```
workpath/common/FTE
```

On the broker system SYSD, we found the workpath to be:

```
C:\Documents and Settings\All Users\Application Data\IBM\MQSI\
```

8.5.3 Creating message flows with FTE nodes

The FTEInput and FTEOutput nodes can be found in the WebSphere Message Broker Toolkit node palette in the File section (Figure 8-45).

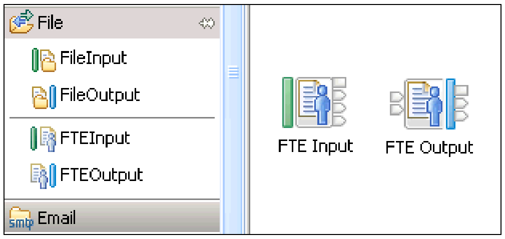


Figure 8-45 WebSphere Message Broker Toolkit - File drawer with FTE nodes

You do not need to configure the WebSphere MQ File Transfer Edition code that runs in the broker. Operational tools in WebSphere Message Broker Explorer are provided to create transfers. At a high-level, these are the steps for using the nodes to send or receive data across an existing WebSphere MQ File Transfer Edition network:

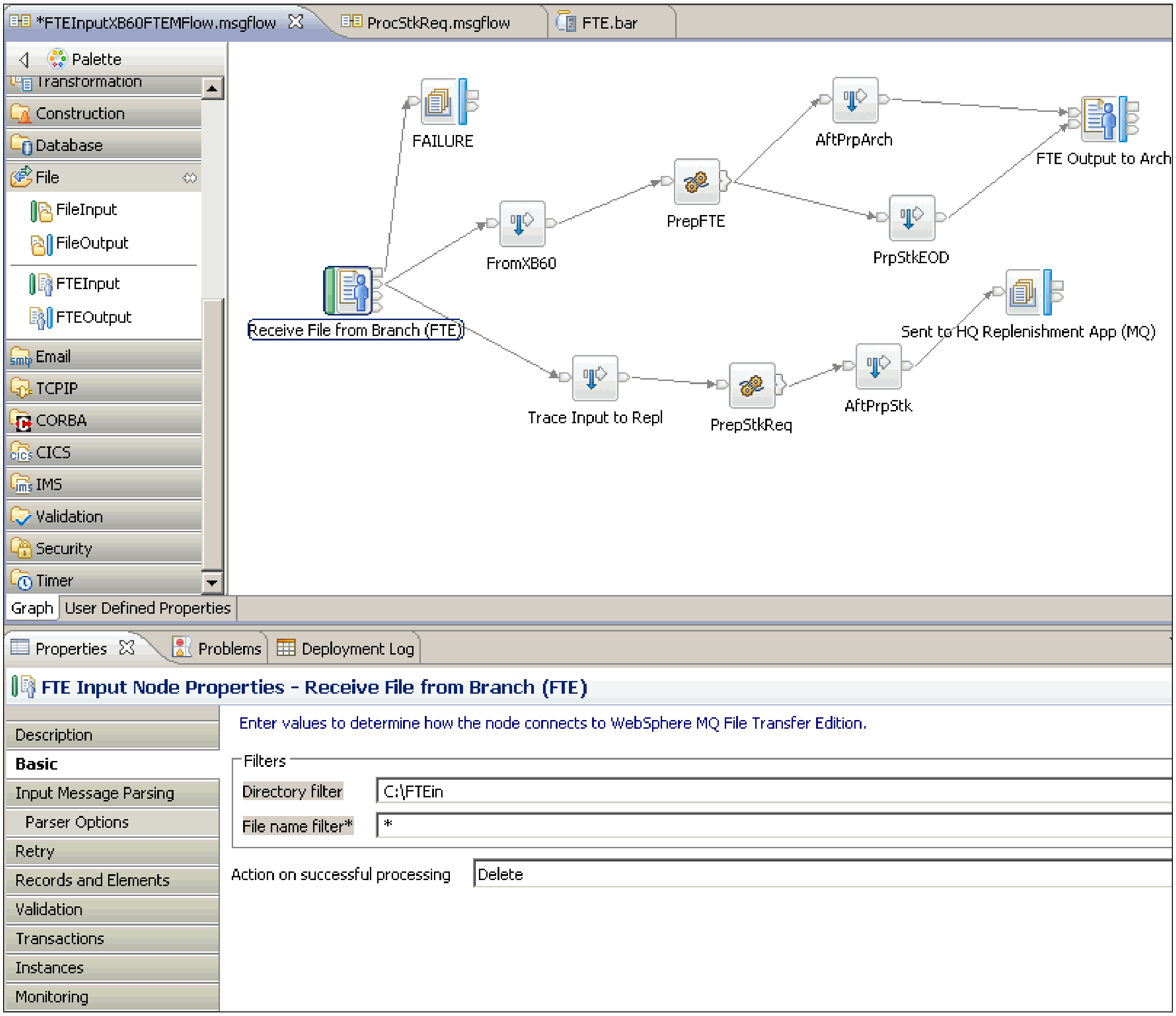
1. Create a flow that includes one of the FTE nodes.
2. Configure the node.
3. For production purposes, change the coordination queue manager from the broker queue manager.
4. Deploy the flow.

Using the FTEInput node

The FTEInput node is used to extend WebSphere Message Broker Version 7.0 support for file processing through its integration with WebSphere MQ File Transfer Edition. Use this node in a flow that expects to receive files from a WebSphere MQ File Transfer Edition agent in the MQ FTE backbone network. The node is configured with properties like any other WebSphere Message Broker input node. The properties tell the node where to find the files to be processed, the file name pattern, and how to parse the data being received. See the WebSphere Message Broker V7.0.0.1 Information Center for tables of the properties settings.

http://publib.boulder.ibm.com/infocenter/wmbhelp/v7r0m0/topic/com.ibm.etools.mft.doc/bc34034_.htm

Notice that the FTEInput node in the message flow in Figure 8-46 is waiting for files to arrive in the `C:\FTEin` directory and is accepting any file name, as indicated by the asterisk. The properties panel also indicates the disposition of the file after it is processed by the FTEInput node. Valid options are no action, add a time stamp, and delete.



FTE Input Node Properties - Receive File from Branch (FTE)

Enter values to determine how the node connects to WebSphere MQ File Transfer Edition.

Filters

Directory filter: C:\FTEin

File name filter*: *

Action on successful processing: Delete

Figure 8-46 Inbound message flow with FTEInput node

The following steps were performed to build the message flow with an FTEInput node for this scenario. The complete flow is shown in Figure 8-46. Detailed information about configuring the FTEInput node is given on the property panels for the node in the WebSphere Message Broker Toolkit.

1. Drag an FTEInput node onto a message flow and wire its out terminal to an output node of your choice. Notice that in our flow there is a trace node after the FTEInput node so that the inbound file can be examined.

   To process the file based on details of the transfer, place a node such as the route node after the FTEInput node.

   Details of the transfer are stored in the local environment, at LocalEnvironment.FTE.

2. Configure the properties of the FTEInput node in the Basic panel (shown in the lower half of Figure 8-46 on page 226). The basic tab indicates the directory that contains the files to be processed by the node. To process only a defined subset of files sent to an agent, configure the file name filter. This action allows multiple FTEInput nodes in the same execution group to receive specific files, depending on the directory or file filters specified.

   When receiving files, you can apply filters. If an execution group has more than one FTEInput node, each node receives only the appropriate files. You can also determine what happens after the file has been processed (the file is left in its existing destination directory, left with a time stamp added, or deleted). See the Basic tab on the node for details.

   Note that the FTEInput node does not use a transit directory like the FileInput node. Each execution group has its own FTE agent, and a node processes only files sent to the FTE agent to which the node is deployed. The execution group ensures that only one node in the execution group processes each file.

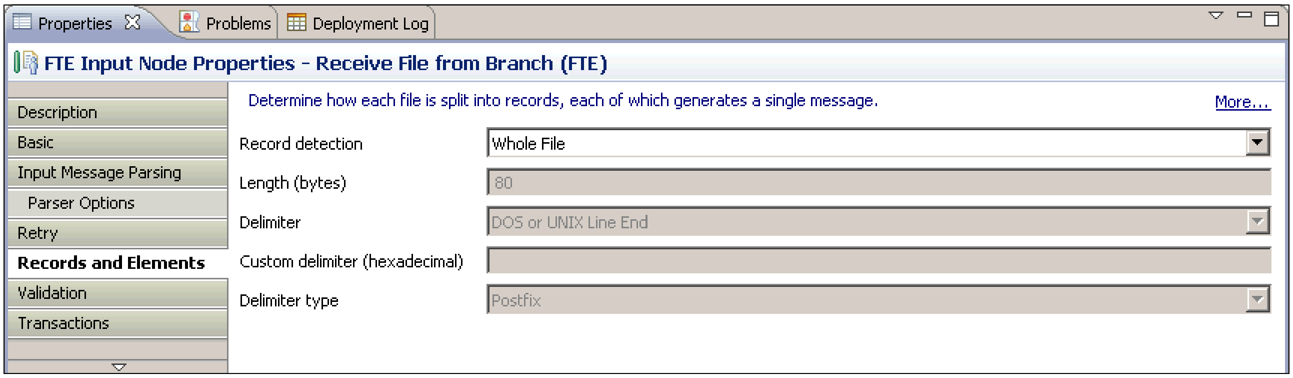
   You can also specify whether, after processing, the file should be left in its directory, renamed, or deleted.

3. To change how the node handles a message flow failure, configure the Retry panel.

4. To change how records are identified in the input file, configure the record detection property on the Records and Elements panel (Figure 8-47). For example, you might want to specify that a record is fixed length and set the record length. Notice that we process the message as a whole file. Use the pull-down menu to select one of the other options:

   – Fixed length
   – Delimited
   – Parsed record sequence

   If you set the record detection property to anything other than Whole File, drag an additional output node to the flow, such as the MQOutput node. Wire the End of data terminal on the FTEInput node to the in terminal of the MQOutput node. The node connected to the end of data terminal receives an empty message when the last record in the file is read.



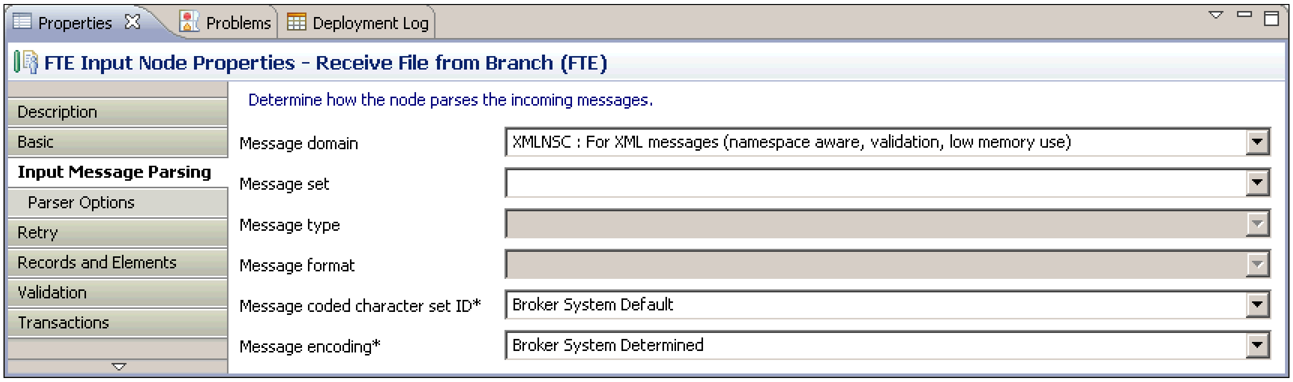
FTE Input Node Properties - Receive File from Branch (FTE)

Determine how each file is split into records, each of which generates a single message.

Record detection: Whole File
Length (bytes): 80
Delimiter: DOS or UNIX Line End
Custom delimiter (hexadecimal):
Delimiter type: Postfix

Figure 8-47 FTEInput node properties panel for records and elements - Message is Whole File

5. The Input Message Parsing property tells how to parse the data. We use XMLNSC parser (Figure 8-48).



FTE Input Node Properties - Receive File from Branch (FTE)

Determine how the node parses the incoming messages.

Message domain: XMLNSC : For XML messages (namespace aware, validation, low memory use)
Message set:
Message type:
Message format:
Message coded character set ID*: Broker System Default
Message encoding*: Broker System Determined

Figure 8-48 FTEInput node properties panel for input message parsing - XMLNSC

6. Add the flow to a broker archive (BAR) file and deploy the BAR file.

Using the FTEOutput node

The FTEOutput node can be used in a message flow that needs to send a file using the MQ FTE backbone network.

Multiple FTEOutput nodes can be deployed to the same execution group or to different execution groups in the same broker. But an FTEOutput node can only send one file per transfer. Each file can have multiple records and each record can have multiple elements.

Transfers from the FTEOutput are non-blocking. An error occurs if another transfer is outstanding with the same file name. This suggests that flows with an FTEOutput node are single threaded if you use the same file name for each occurrence of the flow. If you code the properties to create unique file names, then you should not encounter transfer errors because of duplicate file names.

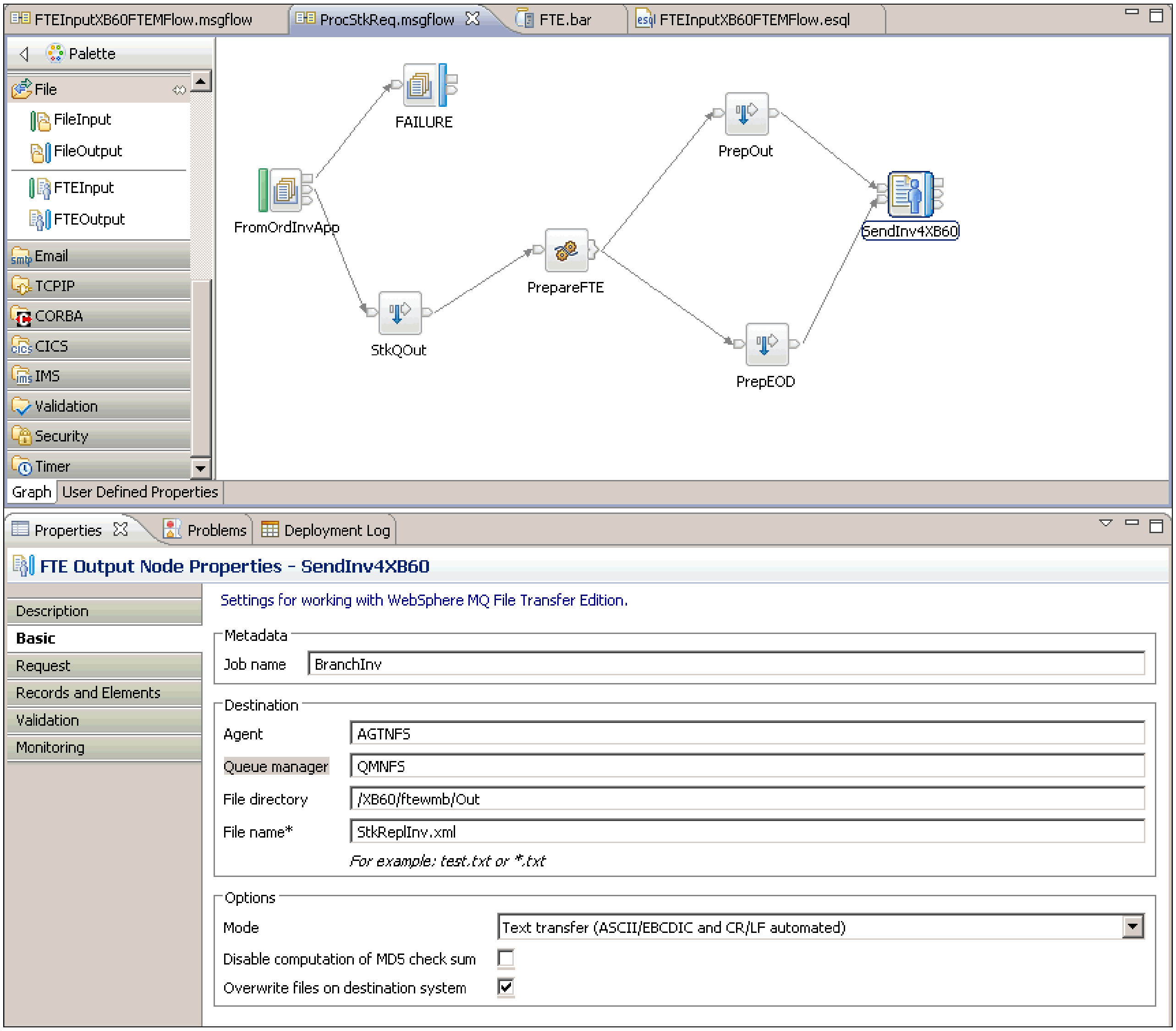
When sending a file, you can dynamically set the following properties:

- Destination agent
- Destination file directory
- Destination file name
- Destination queue manager
- Job name
- Overwrite files on destination

See Figure 8-49 for properties for setting the file options. Anyone who has created FTE transfers will recognize these parameters. They are equivalent to destination parameters in an **`fteCreateTransfer`** request:

- Destination agent name
- Destination agent queue manager
- Destination directory
- Destination file

Also notice the options for mode of transfer and overwriting the destination file.



FTE Output Node Properties - SendInv4XB60

Settings for working with WebSphere MQ File Transfer Edition.

Metadata

Job name: BranchInv

Destination

Agent: AGTNFS

Queue manager: QMNFS

File directory: /XB60/ftewmb/Out

File name*: StkReplInv.xml

*For example: test.txt or *.txt*

Options

Mode: Text transfer (ASCII/EBCDIC and CR/LF automated)

Disable computation of MD5 check sum ☐

Overwrite files on destination system ☑

Figure 8-49 Message with FTEOutput node and corresponding properties

We needed the following information to configure our message flow with a FTEOutput node:

- The name of the remote WebSphere MQ File Transfer Edition agent to which the file is to be sent.
- The name of the destination queue manager.
- The name of the output file.

Using Figure 8-49 on page 229 as reference, the steps below were performed to build the message with a FTEOutput node for use in the outbound part of the scenario. Detailed information about configuring the FTEOutput node is given on the property panels for the node in the WebSphere Message Broker Toolkit. Create a message flow that contains an input node.

1. Drag an FTEOutput node onto the message flow, and wire its in terminal to the input node.
2. Configure the Basic panel (shown in the lower half of Figure 8-49 on page 229):
   a. Set values for the destination agent and destination file name properties. Configuring just these two properties is enough if you want to send all of the input message tree as a single record in the output file.
   b. Set a value for the destination queue manager property. The default destination queue manager is the queue manager for the broker.
3. To specify a location in the input message tree for the data to be sent, configure the data location property on the Request properties panel.
4. To change how records are placed in the output file, configure the record definition property on the Records and Elements panel. For example, you might want to specify that a record is fixed length, and set the record length.

   If you set the record definition property to anything other than Record is Whole File, drag a node such as the MQOutput node to the flow and wire its out terminal to the finish file terminal on the FTEOutput node. The node connected to the finish file terminal must have logic to determine the last record in the file.

   In the flow, notice that we chose to use compute node and handle the end of file logic in the compute and wired the Out1 terminal to the finish file terminal of the FTEOutput node.
5. If required, configure the node to write the overrides to the LocalEnvironment.Destination.FTE subtree (Table 8-6 on page 234). To set properties for the transfer dynamically, place a node such as the compute node or the mapping node before the FTEOutput node. You can override the following properties:
   - Destination agent
   - Destination file directory
   - Destination file name
   - Destination queue manager
   - Job name
   - Overwrite files on destination

   We did not use any overrides in this scenario.
6. Use a node such as the compute node or the mapping node before the FTEOutput node to alter any of the properties or to add or change headers for the FTE transfer.
7. Add the flow to a broker archive (BAR) file and deploy the BAR file.

Local environment tree structure for FTE nodes

The local environment tree is a part of the logical message tree, in which you can store information while the message flow processes the message. The root of the local environment tree is called LocalEnvironment. This tree is always present in the input message. It is created when a message is received by the input node. Certain input nodes create local environment fields. Others leave it empty.

Use the local environment tree to store variables that can be referred to and updated by message processing nodes that occur later in the message flow. You can also use the local environment tree to define destinations (both internal and external to the message flow) to which a message is sent. WebSphere Message Broker also stores information in LocalEnvironment in certain circumstances, and references it to access values that you might have set for destinations. (Compare this with the environment tree structure, which the broker only uses in specific situations.)

Figure 8-50 shows an example of the local environment tree structure. The children of destination are protocol-dependent.

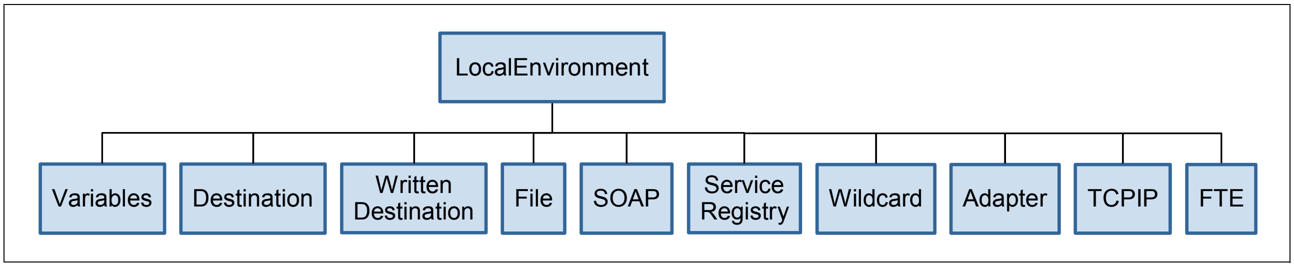


Figure 8-50 Local.Environment tree structure

In the tree structure shown in Figure 8-50, the local environment has several children. This subtree is optional. If you create local environment variables, store them in a subtree called variables. This subtree provides a work area that you can use to pass information between nodes. This subtree is never inspected or modified by any supplied node.

Variables in the local environment can be changed by any subsequent message processing node, and the variables persist until the node that created them goes out of scope. The variables in this subtree are persistent only within a single instance of a message flow. If you have multiple instances of a message passing through the message flow and need to pass information between them, you must use an external database.

You can use fields in the local environment to dynamically alter the behavior of the FTEInput and FTEOutput nodes. You can also find what values the output nodes used to process the file.

LocalEvironment.Wildcard.WildcardMatch tree structure for FTE nodes

On the FileInput and FTEInput nodes, you can specify a file name pattern that contains wildcard characters. The input nodes copy the characters in the file name matched by wildcards, together with any intermediate characters, to LocalEnvironment.Wildcard.WildcardMatch.

Table 8-2 LocalEnvironment.Wildcard.WildcardMatch field

Element name	Element data type	Description
Wildcard match	CHARACTER	You can use fields in the local environment to dynamically alter the behavior of the FileInput, FileOutput, FTEInput, and FTEOutput nodes. You can also find which values the output nodes used to process the file.

LocalEvironment.FTE tree structure for FTE nodes

When you use the FTEInput node, it stores information that you can access in the LocalEnvironment.FTE and LocalEnvironment.FTE.Transfer message trees.

The LocalEnvironment.FTE subtree stores information relating to the current record and is populated by the broker. Table 8-3 describes the fields in this structure.

Table 8-3 LocalEnvironment.FTE fields

Element name	Element data type	Description
TimeStamp	CHARACTER	You can use fields in the local environment to dynamically alter the behavior of the FileInput, FileOutput, FTEInput, and FTEOutput nodes. You can also find which values the output nodes used to process the file.
Offset	INTEGER	Start of the record within the file. The first record starts at offset 0 bytes. When offset is part of the end of data message tree, this value is the length of the input file.
Record	INTEGER	Number of the record within the file. The first record is record number 1. When the record is part of the end of data message tree, this value is the number of records.
Delimiter	CHARACTER	The characters used to separate this record from the preceding record, if delimited is specified in record detection. The first record has a null delimiter. When delimiter is part of the end of data message tree, this value is the delimiter that follows the last record, if any.
IsEmpty	BOOLEAN	Whether the record propagated by the message flow is empty. IsEmpty is set to TRUE if the current record is empty. When IsEmpty is part of the end of data message tree, this value is always set to TRUE.

Local Evironment.FTE.Transfer tree structure for FTE nodes

The LocalEnvironment.FTE.Transfer subtree contains information received from WebSphere MQ File Transfer Edition regarding the transfer or file. Table 8-4 describes the fields in this structure.

Table 8-4 LocalEnvironment.FTE.Transfer message tree

Element name	Element data type	Description
Directory	CHARACTER	The absolute directory path of the input directory.
JobName	CHARACTER	The name of the transfer.
Name	CHARACTER	The file name and extension (per file).
LastModified	TimeStamp	Date and time the file was last modified (per file).
SourceAgent	CHARACTER	The name of the agent sending the file.
DestinationAgent	CHARACTER	The name of the agent to which to send the file.
OriginatingHost	CHARACTER	The name of the host from which the transfer was submitted.
TransferId	CHARACTER	The unique name of the transfer.
MQMDUser	CHARACTER	The MQ user ID in the MQMD of the transfer message.
OriginatingUser	CHARACTER	The user ID of the user that submitted the transfer request.

Element name	Element data type	Description
TransferMode	CHARACTER	The mode of the transfer. Valid values are binary and text.
TransferStatus	CHARACTER	The status of the transfer of the file.
FileSize	INTEGER	The size of the file being transferred.
ChecksumMethod	CHARACTER	The only allowed value is MD5.
Checksum	CHARACTER	If the ChecksumMethod element is set to MD5, this element is the actual checksum in hex string format.
DestinationAgentQmgr	CHARACTER	The name of the destination agent's queue manager to which to send the file.
SourceAgentQmgr	CHARACTER	The name of the source agent's queue manager that sent the file.
OverallTransferStatus	CHARACTER	The overall status of the transfer.
TotalTransfers	INTEGER	The total number of files successfully transferred.
TransferNumber	INTEGER	The number of the current file in the transfer.

The LocalEnvironment.FTE and LocalEnvironment.FTE.Transfer structures are propagated with each message written to the out terminal of the FTEInput node and with the empty message written to the end of data terminal.

Local Evironment.WrittenDestination tree structure for FTE nodes

When you use the FTEOutput node, it stores information that you can access in the LocalEnvironment.WrittenDestination.FTE message tree. Table 8-5 describes the fields in this structure.

Table 8-5 LocalEnvironment.WrittenDestination.FTE fields

Element name	Element data type	Description
DestinationAgent	CHARACTER	The name of the agent to which to send the file.
DestinationAgentQmgr	CHARACTER	The name of the destination queue manager.
JobName	CHARACTER	The name for the transfer.
Directory	CHARACTER	The absolute directory path of the output directory in the form used by the file system of the broker. For example, on Windows systems, this starts with the drive letter prefix (such as C:).
Name	CHARACTER	The file name of the output file.
Overwrite	BOOLEAN	The absolute directory path of the output directory in the form used by the file system of the broker. For example, on Windows systems, this starts with the drive letter prefix (such as C:).

Local Evironment.Destination.FTE tree structure for FTE nodes

This subtree consists of a number of children that indicate the transport types to which the message is directed (the transport identifiers) or the target label nodes that are used by a RouteToLabel node.

Transport information is used by certain input and output nodes, including:

- FTE
- HTTP
- MQ
- JMS
- SOAP
- File
- E-mail
- TCPIP

When you use the FTEOutput node, you can override its destination agent, destination queue manager, job name, destination file directory, destination file name, and overwrite files on destination system properties with elements in the message tree. The default location for these overrides is LocalEnvironment.Destination.FTE. Table 8-6 describes the fields of this structure.

Table 8-6 LocalEnvironment.Destination.FTE fields

Element name	Element data type	Description
DestinationAgent	CHARACTER	The name of the agent to which to send the file.
DestinationAgentQmgr	CHARACTER	The name of the destination queue manager.
Jobname	CHARACTER	The name of the transfer.
Directory	CHARACTER	The absolute directory path of the output directory in the form used by the file system of the broker. For example, on a Windows system, this starts with the drive letter prefix (such as C:).
Name	CHARACTER	The file name of the output file.
Overwrite	BOOLEAN	This specifies whether files on the destination system can be overwritten when the destination agent moves files of the same name there. If the destination agent fails to overwrite the file, the transfer fails and the transfer logs report the failure. The FTEOutput node does not throw or log any errors.
TransferId	CHARACTER	The unique name of the transfer initiated by the FTEOuput node.

8.6 Testing the inbound and outbound flows

In this section, both the inbound and the outbound flow are described. The concept of inbound versus outbound is from the perspective of the internal partner.

For both the partner's B2B gateway and your B2B gateway we use a simulated back-end that communicates over an HTTP connection. For the purposes of this scenario, we send the payload files over the HTTP integration points using an HTTP utility called NetTool from SourceForge.net:

http://sourceforge.net/projects/nettool/

The tools used to view the transaction flows and the contents are documented at each step.

The following scenarios are being tested:

- Inbound flow: receiving an XML document from external partner *partner*
- Outbound flow: sending an XML document to external partner *partner*

Receiving a document from external partner named partner (inbound flow)

In the inbound flow, the external partner starts the flow by sending an XML file over SFTP. The document represents a stock replenishment request. The following flow explains how the tests were executed:

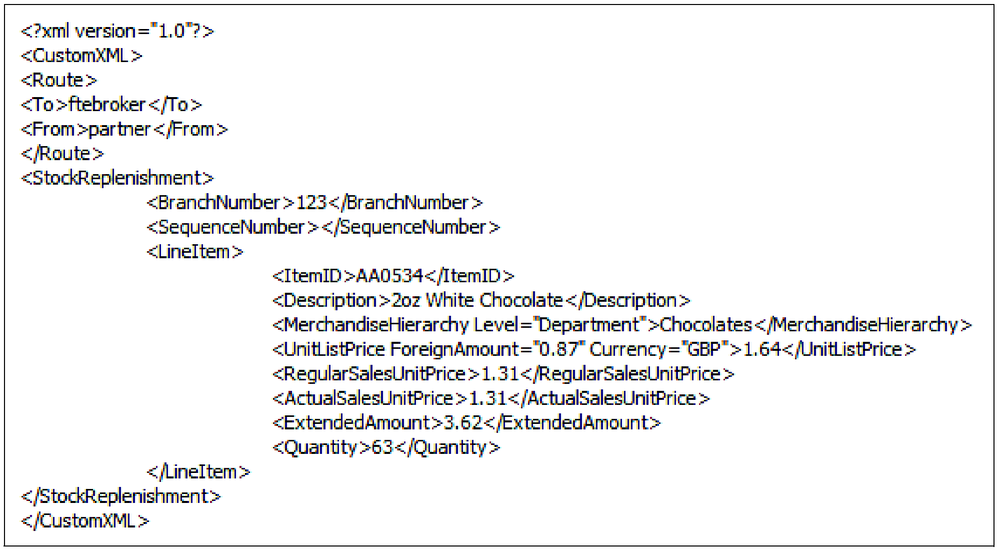
1. The processing starts when the external partner sends a request via the SFTP protocol to the SFTP server front side handler in the XB60 FTE_B2B_GW gateway service. In our test environment, we simulated this action by using NetTool to send the request via HTTP to the HTTP front side handler in the PARTNERHUB B2B gateway.

   Additional materials: The PARTNERHUB gateway is set up on our XB60 to simulate the external partner. Whereas we do not show the configuration of this gateway, we have exported the PARTNER domain and included it in the Common_Files directory in Appendix C, “Additional material” on page 317. If you import this domain, you must import the certificates for the PARTNER and B2BFTE profiles after importing the domain. The certificates can be found in the B2BScenario_Files\certs folder in Appendix C, “Additional material” on page 317.

   The port for the HTTP front side handler in the PARTNERHUB gateway is 30003.

   The routing information in the request indicates that the request is to ftebroker and from partner, so that the file is routed to the broker. These business IDs are defined as partners in the B2B gateway.

Figure 8-51 shows the file that is transmitted.



```
<?xml version="1.0"?>
<CustomXML>
<Route>
<To>ftebroker</To>
<From>partner</From>
</Route>
<StockReplenishment>
                <BranchNumber>123</BranchNumber>
                <SequenceNumber></SequenceNumber>
                <LineItem>
                                <ItemID>AA0534</ItemID>
                                <Description>2oz White Chocolate</Description>
                                <MerchandiseHierarchy Level="Department">Chocolates</MerchandiseHierarchy>
                                <UnitListPrice ForeignAmount="0.87" Currency="GBP">1.64</UnitListPrice>
                                <RegularSalesUnitPrice>1.31</RegularSalesUnitPrice>
                                <ActualSalesUnitPrice>1.31</ActualSalesUnitPrice>
                                <ExtendedAmount>3.62</ExtendedAmount>
                                <Quantity>63</Quantity>
                </LineItem>
</StockReplenishment>
</CustomXML>
```

Figure 8-51 Stock replenishment request being sent from NetTool

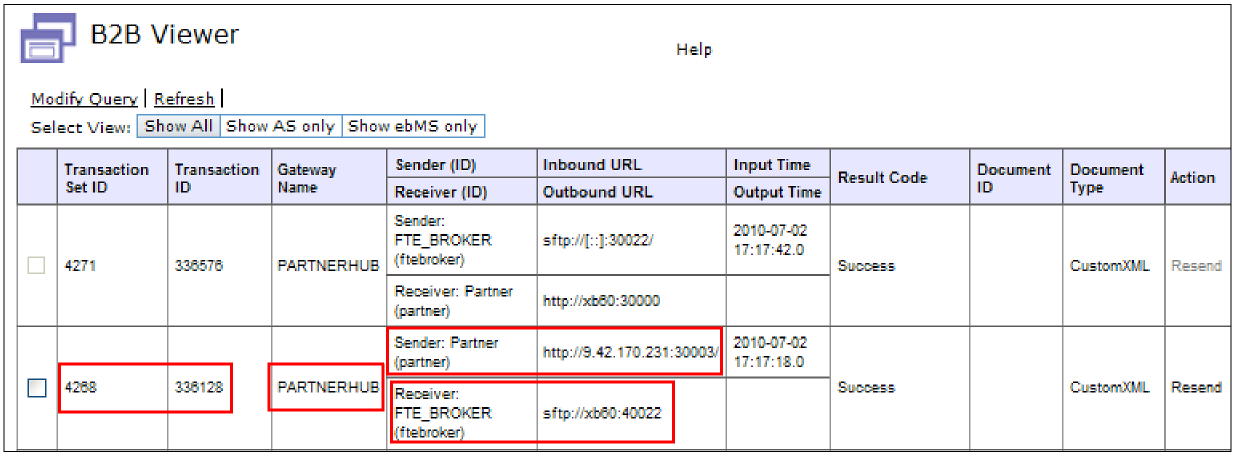
2. The B2B Gateway service of PARTNERHUB looks up the partner information and verifies that the partners exist and are allowed to trade documents.

   The gateway looks at the destination that is configured for the receiving profile (FTE_BROKER) and delivers the payload to the appropriate address.

3. Now let us view the transaction in the XB60's B2B transaction viewer. Click **Control Panel** and then **B2B Transaction Viewer**.

   The partner's B2B gateway parses the XML file and extracts the sender (<From>) and receiver (<To>) information from the file. Figure 8-51 shows where the sender and receiver information is located in the XML file.

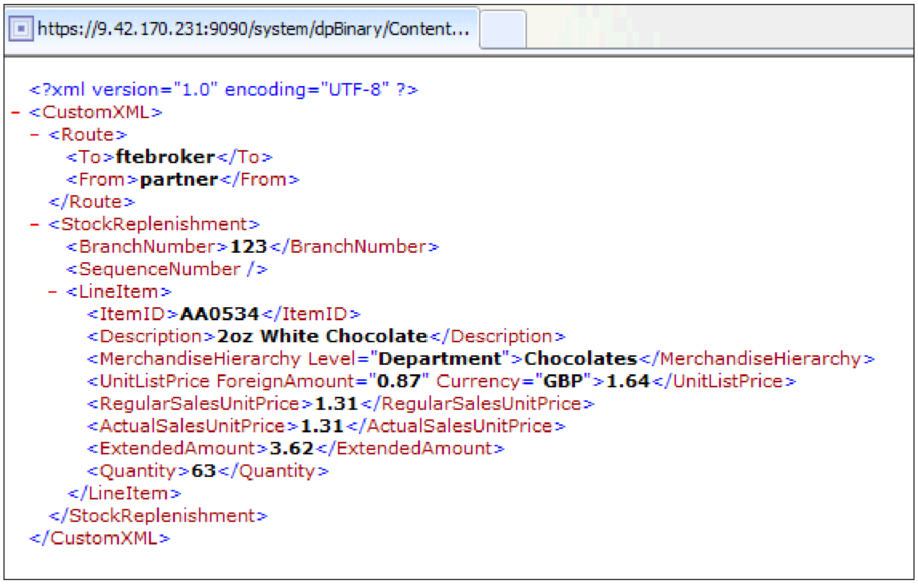
The transaction is found in the B2B transaction viewer for the PARTNERHUB B2B gateway (Figure 8-52). By looking at the entry for Transaction Set ID 4268, we see that the PARTNERHUB gateway received the request via the HTTP front side handler at the `http://9.42.17.231:30003` address and sent it to the FTE_BROKER partner via the SFTP protocol to the SFTP server front side handler in the FTE_B2B_GW listening on port 40022. Transaction Set ID 4271 is the response message to the request in Transaction Set ID 4268.



	Transaction Set ID	Transaction ID	Gateway Name	Sender (ID)	Inbound URL	Input Time	Result Code	Document ID	Document Type	Action
				Receiver (ID)	Outbound URL	Output Time				
□	4271	336576	PARTNERHUB	Sender: FTE_BROKER (ftebroker)	sftp://[::]:30022/	2010-07-02 17:17:42.0	Success		CustomXML	Resend
				Receiver: Partner (partner)	http://xb60:30000					
□	4268	336128	PARTNERHUB	Sender: Partner (partner)	http://9.42.170.231:30003/	2010-07-02 17:17:18.0	Success		CustomXML	Resend
				Receiver: FTE_BROKER (ftebroker)	sftp://xb60:40022					

Figure 8-52 Transaction viewer for PARTNERHUB gateway

Click the Transaction Set ID (4268) and select **Content** to display the data (Figure 8-53). This matches the data that was sent (Figure 8-51 on page 236).

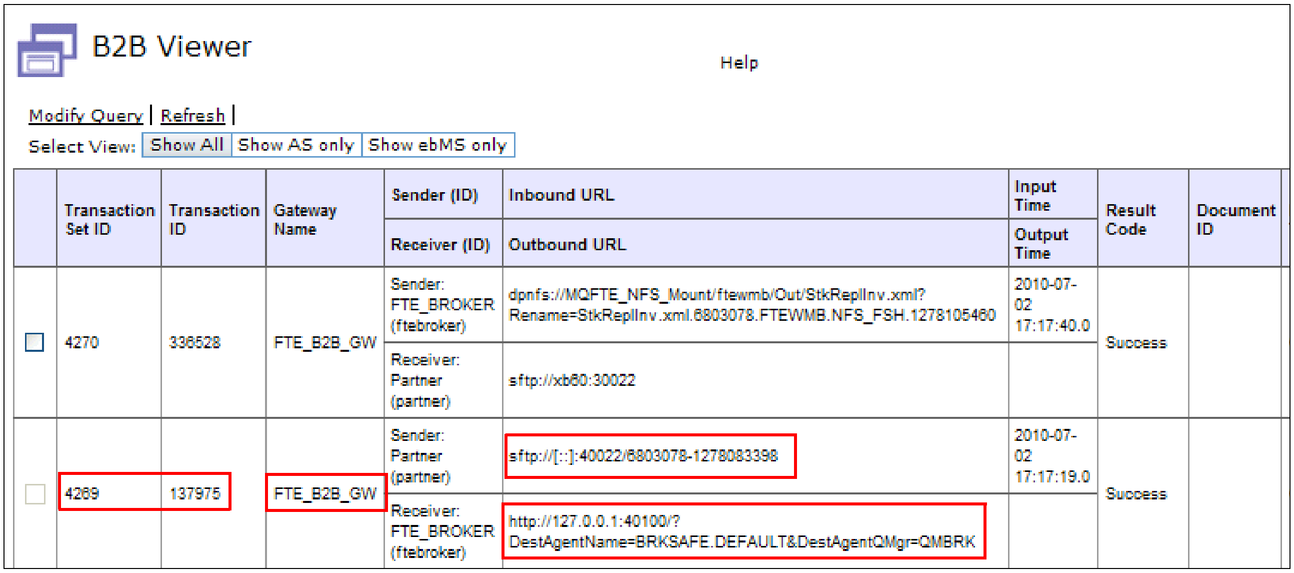


https://9.42.170.231:9090/system/dpBinary/Content...

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <CustomXML>
  - <Route>
      <To>ftebroker</To>
      <From>partner</From>
    </Route>
  - <StockReplenishment>
      <BranchNumber>123</BranchNumber>
      <SequenceNumber />
    - <LineItem>
        <ItemID>AA0534</ItemID>
        <Description>2oz White Chocolate</Description>
        <MerchandiseHierarchy Level="Department">Chocolates</MerchandiseHierarchy>
        <UnitListPrice ForeignAmount="0.87" Currency="GBP">1.64</UnitListPrice>
        <RegularSalesUnitPrice>1.31</RegularSalesUnitPrice>
        <ActualSalesUnitPrice>1.31</ActualSalesUnitPrice>
        <ExtendedAmount>3.62</ExtendedAmount>
        <Quantity>63</Quantity>
      </LineItem>
    </StockReplenishment>
  </CustomXML>
```

Figure 8-53   File received from partner via SFTP

4. The PARTNERHUB B2B gateway sends the XML file to the company's FTE_B2B_GW B2B gateway. The company's B2B gateway receives the XML message at the SFTP Front-side Handler and extracts the sender and receiver information from the route XML tags. Figure 8-54 shows the entry in the transaction viewer for the FTE_B2B_GW.



B2B Viewer

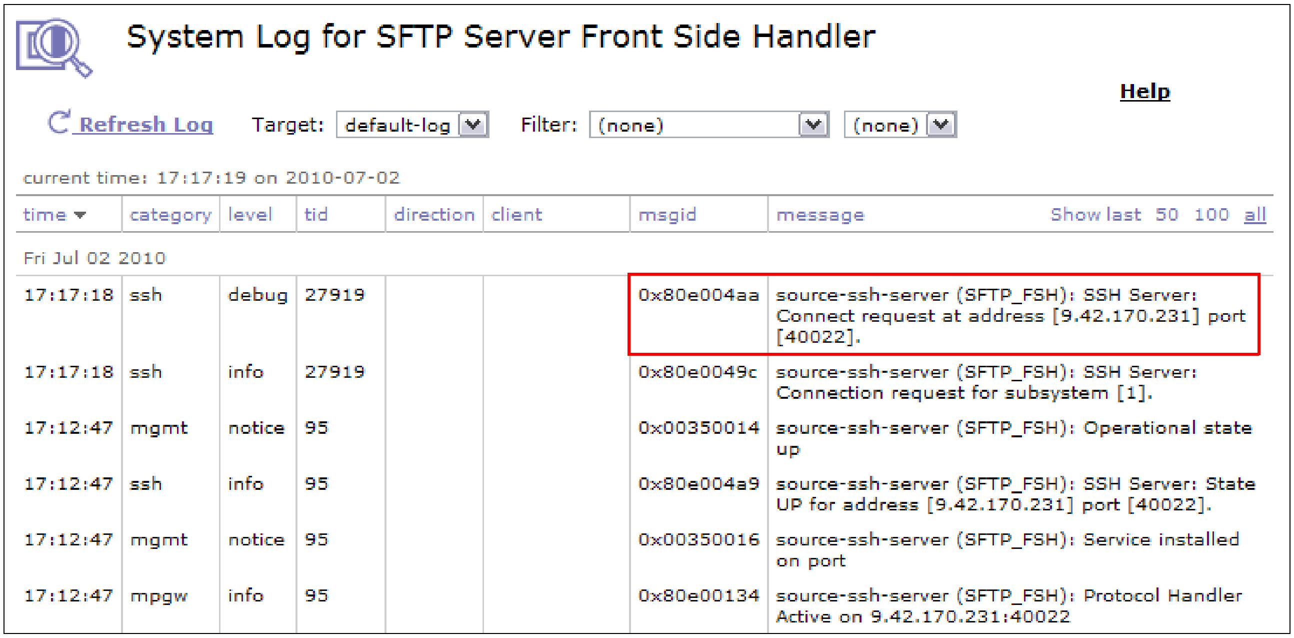
	Transaction Set ID	Transaction ID	Gateway Name	Sender (ID)	Inbound URL	Input Time	Result Code	Document ID
				Receiver (ID)	Outbound URL	Output Time		
☐	4270	336528	FTE_B2B_GW	Sender: FTE_BROKER (ftebroker)	dpnfs://MQFTE_NFS_Mount/ftewmb/Out/StkReplInv.xml?Rename=StkReplInv.xml.6803078.FTEWMB.NFS_FSH.1278105460	2010-07-02 17:17:40.0	Success	
				Receiver: Partner (partner)	sftp://xb60:30022			
☐	4269	137975	FTE_B2B_GW	Sender: Partner (partner)	sftp://[::]:40022/6803078-1278083398	2010-07-02 17:17:19.0	Success	
				Receiver: FTE_BROKER (ftebroker)	http://127.0.0.1:40100/?DestAgentName=BRKSAFE.DEFAULT&DestAgentQMgr=QMBRK			

Figure 8-54   Transaction viewer for FTE_B2B_GW gateway

Transaction Set ID 4269, Transaction ID 336128 shows that the message was received via the SFTP front side handler on port 40022 from PARTNER.

You can see more about this transaction in the system log for the front side handler (Figure 8-55).

Viewing the system log: To view the system log, select **View Logs** in the Control Panel. When the log opens, you can filter on an object (for example, the SFTP_FSH front side handler) by clicking its name in a message in the log.



System Log for SFTP Server Front Side Handler

Target: default-log Filter: (none) (none)

current time: 17:17:19 on 2010-07-02

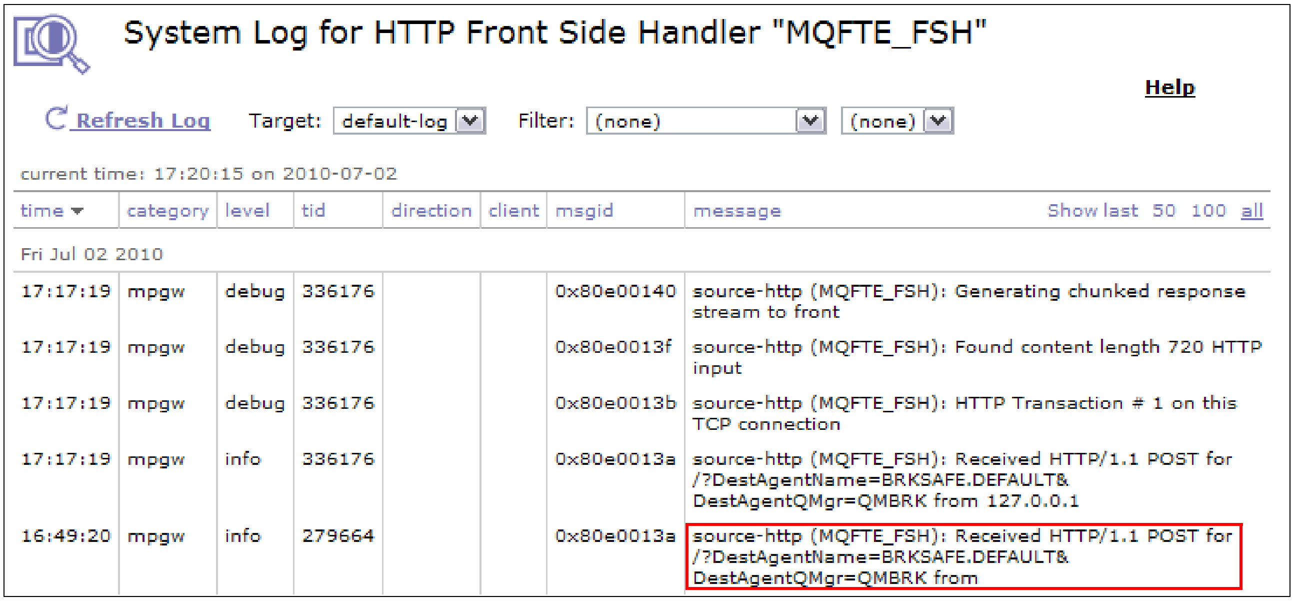
time	category	level	tid	direction	client	msgid	message
Fri Jul 02 2010							
17:17:18	ssh	debug	27919			0x80e004aa	source-ssh-server (SFTP_FSH): SSH Server: Connect request at address [9.42.170.231] port [40022].
17:17:18	ssh	info	27919			0x80e0049c	source-ssh-server (SFTP_FSH): SSH Server: Connection request for subsystem [1].
17:12:47	mgmt	notice	95			0x00350014	source-ssh-server (SFTP_FSH): Operational state up
17:12:47	ssh	info	95			0x80e004a9	source-ssh-server (SFTP_FSH): SSH Server: State UP for address [9.42.170.231] port [40022].
17:12:47	mgmt	notice	95			0x00350016	source-ssh-server (SFTP_FSH): Service installed on port
17:12:47	mpgw	info	95			0x80e00134	source-ssh-server (SFTP_FSH): Protocol Handler Active on 9.42.170.231:40022

Figure 8-55 SFTP request received by SFTP server front side handler

5. When the multi-protocol gateway MQFTE_INTEGRATION receives the file, it writes the file to the shared NFS file system.

   The system log for the HTTP front side handler MQFTE_FSH (Figure 8-56 on page 239) shows that the HTTP request was received and was to be routed to the destination agent BRKSAFE.DEFAULT on queue manager QMBRK, which is running the broker execution group *default*.

Detailed information about the transaction can be found in the SFTP front side handler log (Figure 8-55 on page 238).



System Log for HTTP Front Side Handler "MQFTE_FSH"

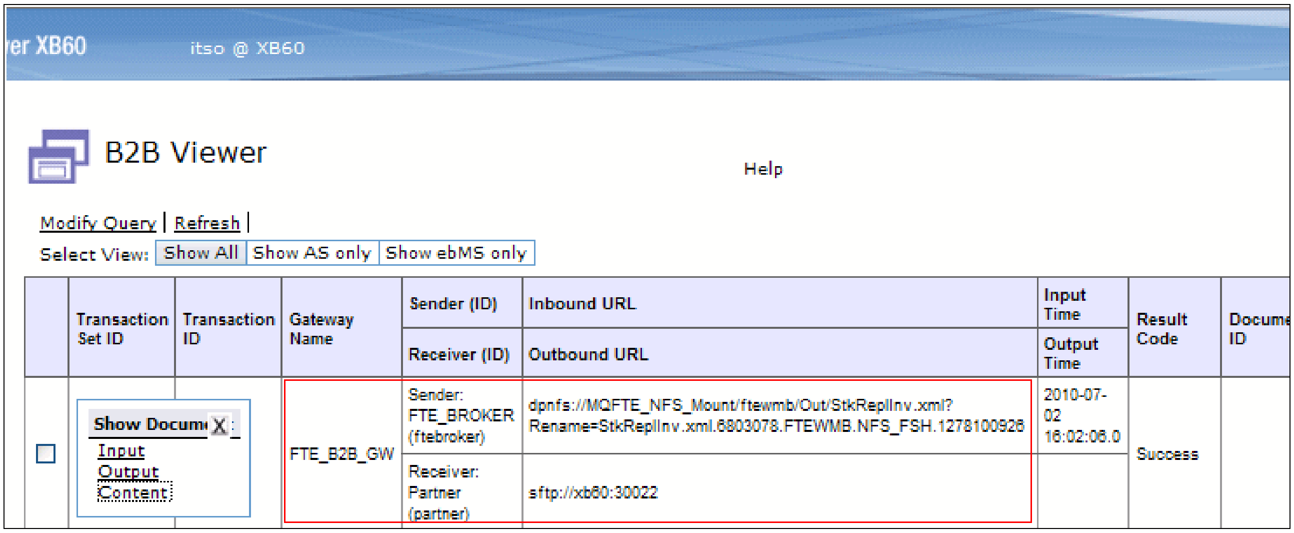
current time: 17:20:15 on 2010-07-02

time	category	level	tid	direction	client	msgid	message
Fri Jul 02 2010							
17:17:19	mpgw	debug	336176			0x80e00140	source-http (MQFTE_FSH): Generating chunked response stream to front
17:17:19	mpgw	debug	336176			0x80e0013f	source-http (MQFTE_FSH): Found content length 720 HTTP input
17:17:19	mpgw	debug	336176			0x80e0013b	source-http (MQFTE_FSH): HTTP Transaction # 1 on this TCP connection
17:17:19	mpgw	info	336176			0x80e0013a	source-http (MQFTE_FSH): Received HTTP/1.1 POST for /?DestAgentName=BRKSAFE.DEFAULT&DestAgentQMgr=QMBRK from 127.0.0.1
16:49:20	mpgw	info	279664			0x80e0013a	source-http (MQFTE_FSH): Received HTTP/1.1 POST for /?DestAgentName=BRKSAFE.DEFAULT&DestAgentQMgr=QMBRK from

Figure 8-56 HTTP FSH in multi-protocol gateway log

The Transaction Set ID 4270 is the response message.

6. The file sent via SFTP is written to the NFS shared file system by the multi-protocol gateway. It is the HTTP front side handler in the FTE_B2B_GW gateway that writes the file. This action can be seen in the B2B transaction viewer (Figure 8-57).



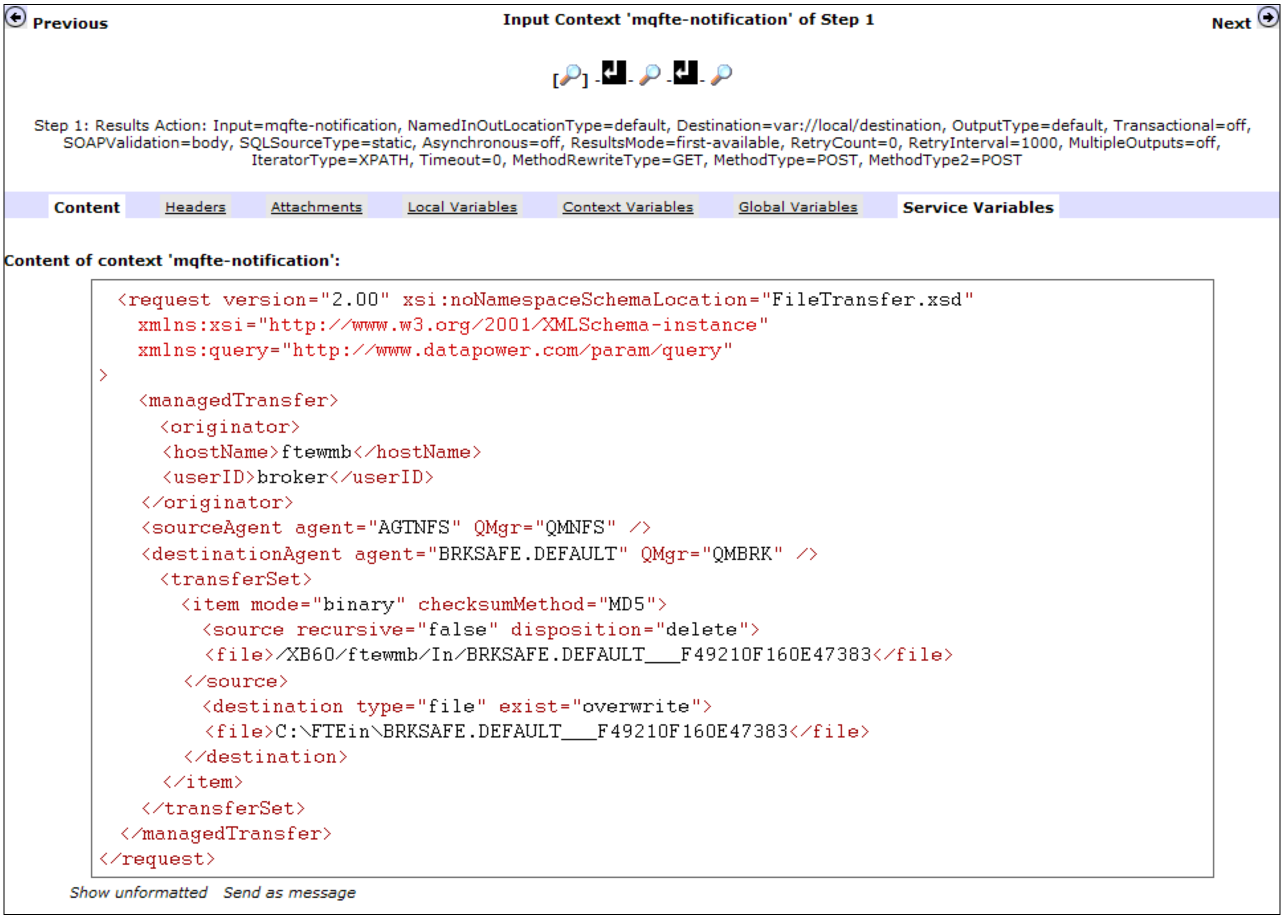
B2B Viewer

	Transaction Set ID	Transaction ID	Gateway Name	Sender (ID) Receiver (ID)	Inbound URL Outbound URL	Input Time Output Time	Result Code	Docume ID
□	Show Docum X Input Output Content		FTE_B2B_GW	Sender: FTE_BROKER (ftebroker)	dpnfs://MQFTE_NFS_Mount/ftewmb/Out/StkReplInv.xml?Rename=StkReplInv.xml.6803078.FTEWMB.NFS_FSH.1278100926	2010-07-02 16:02:06.0	Success	
				Receiver: Partner (partner)	sftp://xb60:30022			

Figure 8-57 Transaction viewer - File written to NFS

The request message is written as a file on the shared NFS files server on SYSE, which is in the protected network behind the firewall. One port must be open in the internal firewall for the XB60 to write the file on the file system.

After the file is written to disk on the file server on SYSE, a command message is put on the command queue of the AGTNFS agent. Figure 8-58 on page 240 shows this message. Notice the directory path and file name on the NFS mount point. Also notice that the source agent is AGTNFS and the destination agent is BRKSAFE.DEFAULT, the agent running in the *default* execution group of broker BRKSAFE.



Input Context 'mqfte-notification' of Step 1

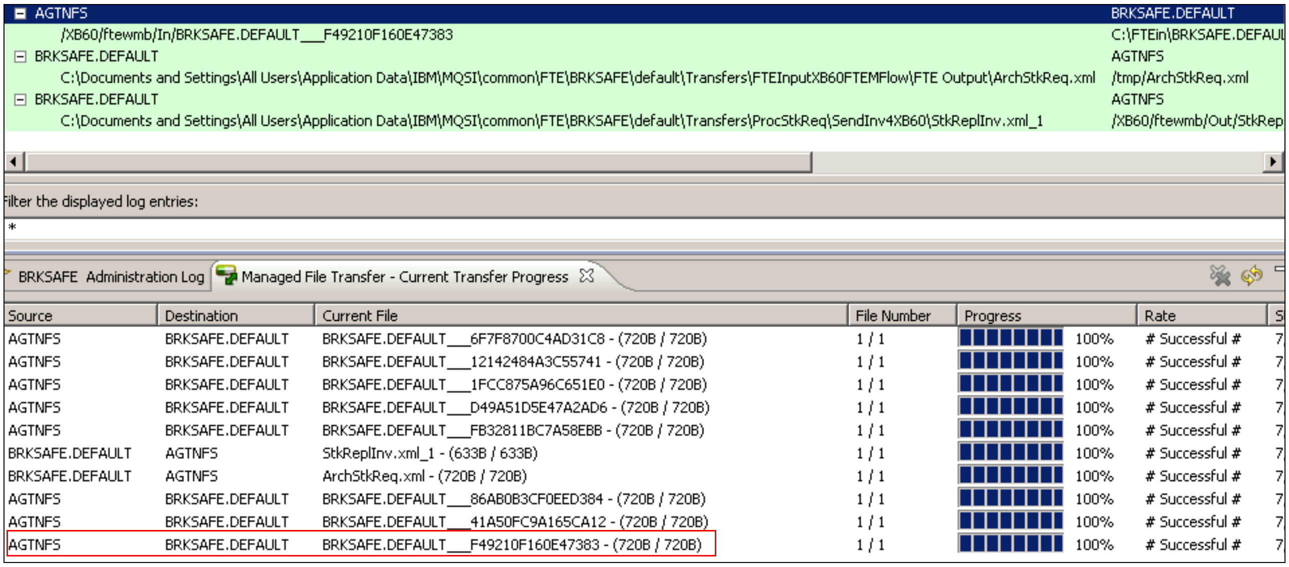
Step 1: Results Action: Input=mqfte-notification, NamedInOutLocationType=default, Destination=var://local/destination, OutputType=default, Transactional=off, SOAPValidation=body, SQLSourceType=static, Asynchronous=off, ResultsMode=first-available, RetryCount=0, RetryInterval=1000, MultipleOutputs=off, IteratorType=XPATH, Timeout=0, MethodRewriteType=GET, MethodType=POST, MethodType2=POST

Content of context 'mqfte-notification':

```
 <request version="2.00" xsi:noNamespaceSchemaLocation="FileTransfer.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:query="http://www.datapower.com/param/query"
>
   <managedTransfer>
     <originator>
     <hostName>ftewmb</hostName>
     <userID>broker</userID>
   </originator>
   <sourceAgent agent="AGTNFS" QMgr="QMNFS" />
   <destinationAgent agent="BRKSAFE.DEFAULT" QMgr="QMBRK" />
     <transferSet>
       <item mode="binary" checksumMethod="MD5">
         <source recursive="false" disposition="delete">
         <file>/XB60/ftewmb/In/BRKSAFE.DEFAULT___F49210F160E47383</file>
       </source>
         <destination type="file" exist="overwrite">
         <file>C:\FTEin\BRKSAFE.DEFAULT___F49210F160E47383</file>
       </destination>
     </item>
   </transferSet>
  </managedTransfer>
</request>
```

Figure 8-58   FTE command message put on AGTNFS command queue by XB60 NFS front side handler

7. AGTNFS sends the file to BRKSAFE.DEFAULT, which writes the file to the `C:\FTEin` directory and initiates the flow. Figure 8-59 shows the file name.



```
⊟ AGTNFS                                                                                                                                    BRKSAFE.DEFAULT
      /XB60/ftewmb/In/BRKSAFE.DEFAULT___F49210F160E47383                                                                                   C:\FTEin\BRKSAFE.DEFAUL
⊟ BRKSAFE.DEFAULT                                                                                                                           AGTNFS
      C:\Documents and Settings\All Users\Application Data\IBM\MQSI\common\FTE\BRKSAFE\default\Transfers\FTEInputXB60FTEMFlow\FTE Output\ArchStkReq.xml   /tmp/ArchStkReq.xml
⊟ BRKSAFE.DEFAULT                                                                                                                           AGTNFS
      C:\Documents and Settings\All Users\Application Data\IBM\MQSI\common\FTE\BRKSAFE\default\Transfers\ProcStkReq\SendInv4XB60\StkReplInv.xml_1   /XB60/ftewmb/Out/StkRep
```

ilter the displayed log entries:

*

BRKSAFE Administration Log | Managed File Transfer - Current Transfer Progress

Source	Destination	Current File	File Number	Progress	Rate	S
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___6F7F8700C4AD31C8 - (720B / 720B)	1 / 1	100%	# Successful #	7
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___12142484A3C55741 - (720B / 720B)	1 / 1	100%	# Successful #	7
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___1FCC875A96C651E0 - (720B / 720B)	1 / 1	100%	# Successful #	7
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___D49A51D5E47A2AD6 - (720B / 720B)	1 / 1	100%	# Successful #	7
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___FB32811BC7A58EBB - (720B / 720B)	1 / 1	100%	# Successful #	7
BRKSAFE.DEFAULT	AGTNFS	StkReplInv.xml_1 - (633B / 633B)	1 / 1	100%	# Successful #	7
BRKSAFE.DEFAULT	AGTNFS	ArchStkReq.xml - (720B / 720B)	1 / 1	100%	# Successful #	7
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___86AB0B3CF0EED384 - (720B / 720B)	1 / 1	100%	# Successful #	7
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___41A50FC9A165CA12 - (720B / 720B)	1 / 1	100%	# Successful #	7
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___F49210F160E47383 - (720B / 720B)	1 / 1	100%	# Successful #	7

Figure 8-59 WebSphere MQ Explorer with WebSphere MQ File Transfer Edition Explorer plug-in transfer log

8. Referring to the message flow in Figure 8-60 and the properties of the FTEInput node in Figure 8-61 on page 242, we see that the FTEInput node "Receive File from Branch (FTE)" is waiting for a file to arrive in the `C:\FTEin` directory. Note that the OUT terminal has two branches. We first address the branch that ultimately leads to the Sent to HQ Replenishment App (MQ) node (the lower branch)

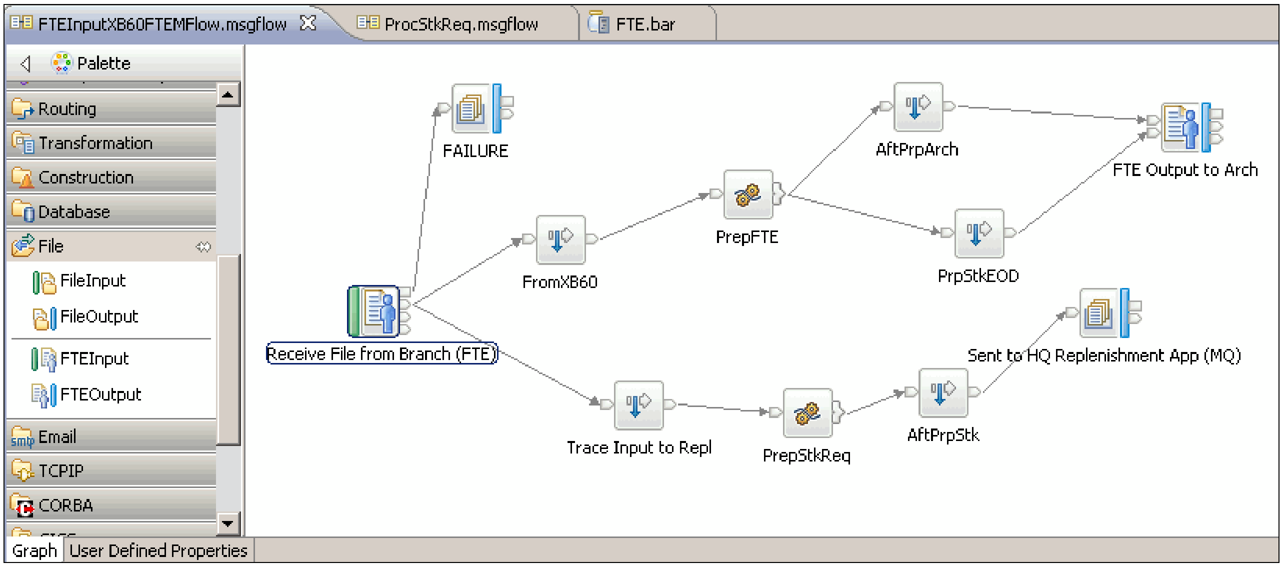
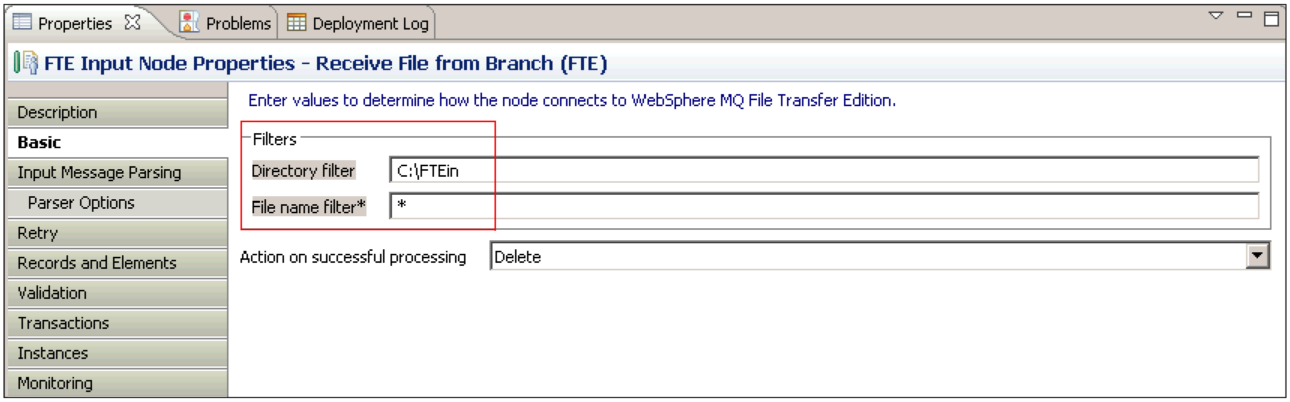


Figure 8-60 Inbound message flow



FTE Input Node Properties - Receive File from Branch (FTE)

Enter values to determine how the node connects to WebSphere MQ File Transfer Edition.

Filters

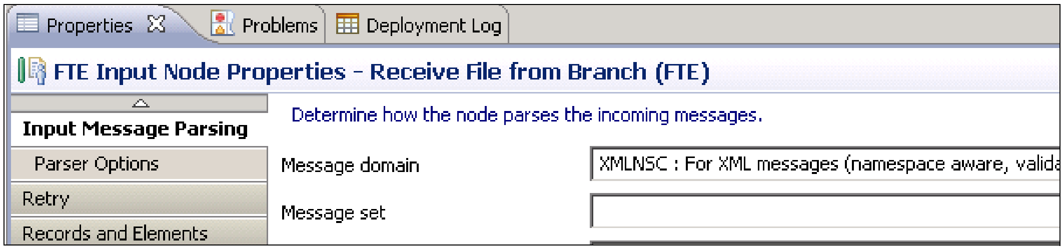
Directory filter: C:\FTEin

File name filter*: *

Action on successful processing: Delete

Figure 8-61 Properties of FTEInput node

The name of the file is not important in this particular message flow, but the Input message parsing properties of the input node (Figure 8-61) tell us that an XML file is expected, because the XMLNSC parser is being used.



FTE Input Node Properties - Receive File from Branch (FTE)

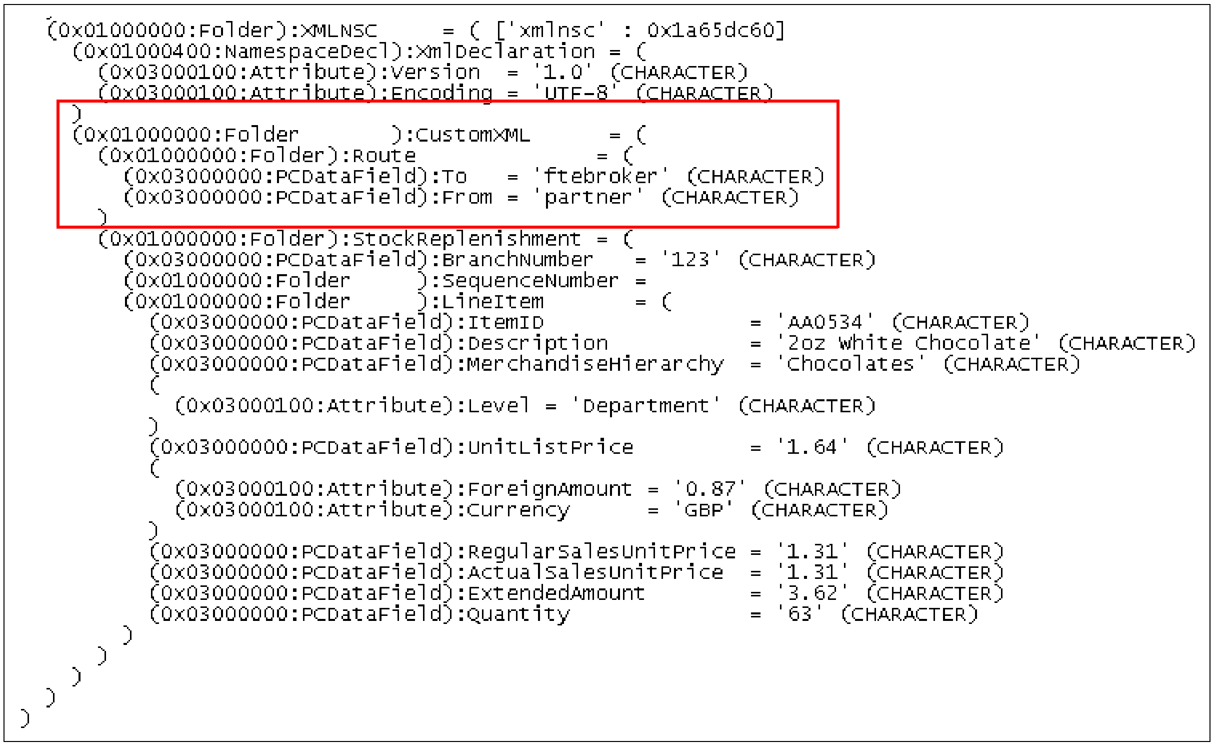
Determine how the node parses the incoming messages.

Message domain: XMLNSC : For XML messages (namespace aware, valida

Message set:

Figure 8-62 Input Message Parsing properties

9. Figure 8-63 shows what the file looks like after it has been received by the BRKSAFE.DEFAULT agent and written to the `C:\FTEin` directory. The display of this file was taken from the Trace Input to Repl trace node.

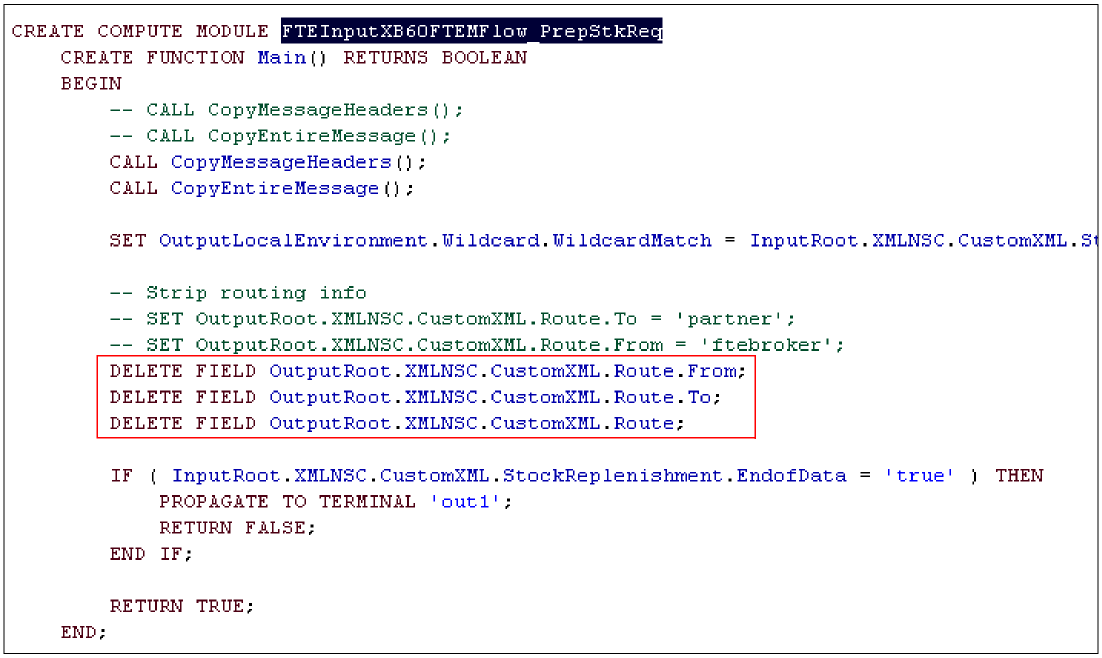


```
 (0x01000000:Folder):XMLNSC     = ( ['xmlnsc' : 0x1a65dc60]
   (0x01000400:NamespaceDecl):XmlDeclaration = (
     (0x03000100:Attribute):Version  = '1.0' (CHARACTER)
     (0x03000100:Attribute):Encoding = 'UTF-8' (CHARACTER)
   )
   (0x01000000:Folder       ):CustomXML      = (
     (0x01000000:Folder):Route                 = (
       (0x03000000:PCDataField):To   = 'ftebroker' (CHARACTER)
       (0x03000000:PCDataField):From = 'partner' (CHARACTER)
     )
     (0x01000000:Folder):StockReplenishment = (
       (0x03000000:PCDataField):BranchNumber   = '123' (CHARACTER)
       (0x01000000:Folder     ):SequenceNumber =
       (0x01000000:Folder     ):LineItem       = (
         (0x03000000:PCDataField):ItemID                 = 'AA0534' (CHARACTER)
         (0x03000000:PCDataField):Description            = '2oz White Chocolate' (CHARACTER)
         (0x03000000:PCDataField):MerchandiseHierarchy  = 'Chocolates' (CHARACTER)
         (
           (0x03000100:Attribute):Level = 'Department' (CHARACTER)
         )
         (0x03000000:PCDataField):UnitListPrice         = '1.64' (CHARACTER)
         (
           (0x03000100:Attribute):ForeignAmount = '0.87' (CHARACTER)
           (0x03000100:Attribute):Currency      = 'GBP' (CHARACTER)
         )
         (0x03000000:PCDataField):RegularSalesUnitPrice = '1.31' (CHARACTER)
         (0x03000000:PCDataField):ActualSalesUnitPrice  = '1.31' (CHARACTER)
         (0x03000000:PCDataField):ExtendedAmount        = '3.62' (CHARACTER)
         (0x03000000:PCDataField):Quantity              = '63' (CHARACTER)
       )
     )
   )
 )
)
```

Figure 8-63 Trace data from Trace Input to Repl node

Notice that the routing information in the <*Route*></Route tags is still in the file. This is required when using profile management, but the stock replenishment (order inventory) application is not concerned with the routing information. It only needs the data between the <*StockReplenishment*></*StockReplenishment*> tags. The message flow must remove the routing information before sending to the application. This is accomplished in the compute node PrepStkReq in the flow.

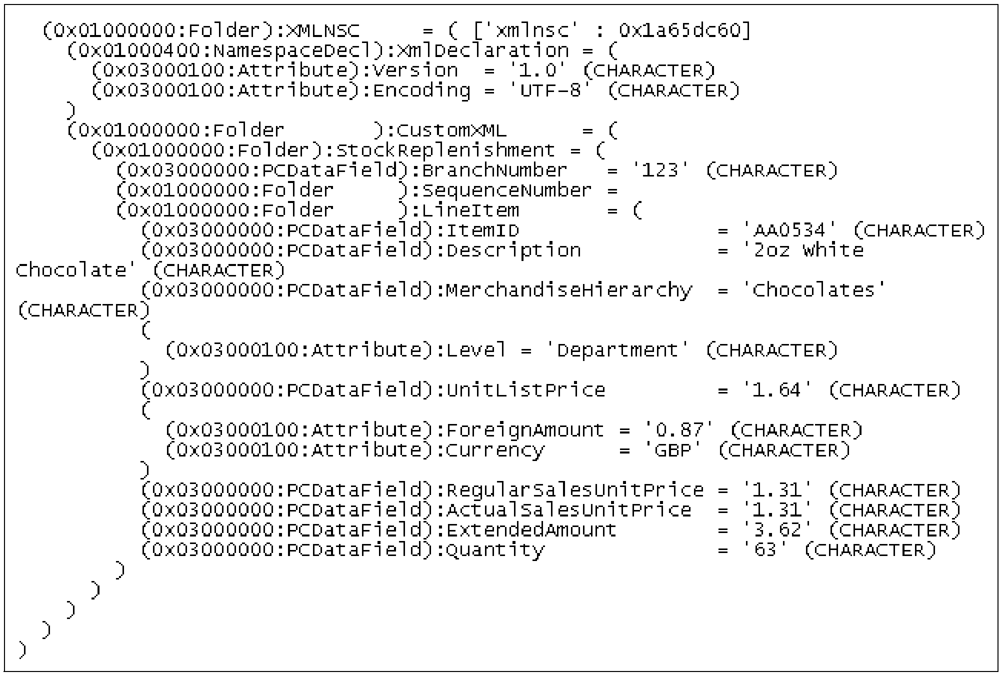
10.The PrepStkReq compute node contains the ESQL code that strips the routing information. Figure 8-64 shows the ESQL code. Notice that it is the delete field statements that is deleting the XML tags from the file.



```
CREATE COMPUTE MODULE FTEInputXB6OFTEMFlow_PrepStkReq
    CREATE FUNCTION Main() RETURNS BOOLEAN
    BEGIN
        -- CALL CopyMessageHeaders();
        -- CALL CopyEntireMessage();
        CALL CopyMessageHeaders();
        CALL CopyEntireMessage();

        SET OutputLocalEnvironment.Wildcard.WildcardMatch = InputRoot.XMLNSC.CustomXML.St

        -- Strip routing info
        -- SET OutputRoot.XMLNSC.CustomXML.Route.To = 'partner';
        -- SET OutputRoot.XMLNSC.CustomXML.Route.From = 'ftebroker';
        DELETE FIELD OutputRoot.XMLNSC.CustomXML.Route.From;
        DELETE FIELD OutputRoot.XMLNSC.CustomXML.Route.To;
        DELETE FIELD OutputRoot.XMLNSC.CustomXML.Route;

        IF ( InputRoot.XMLNSC.CustomXML.StockReplenishment.EndofData = 'true' ) THEN
            PROPAGATE TO TERMINAL 'out1';
            RETURN FALSE;
        END IF;

        RETURN TRUE;
    END;
```

Figure 8-64   ESQL sample code to remove routing information

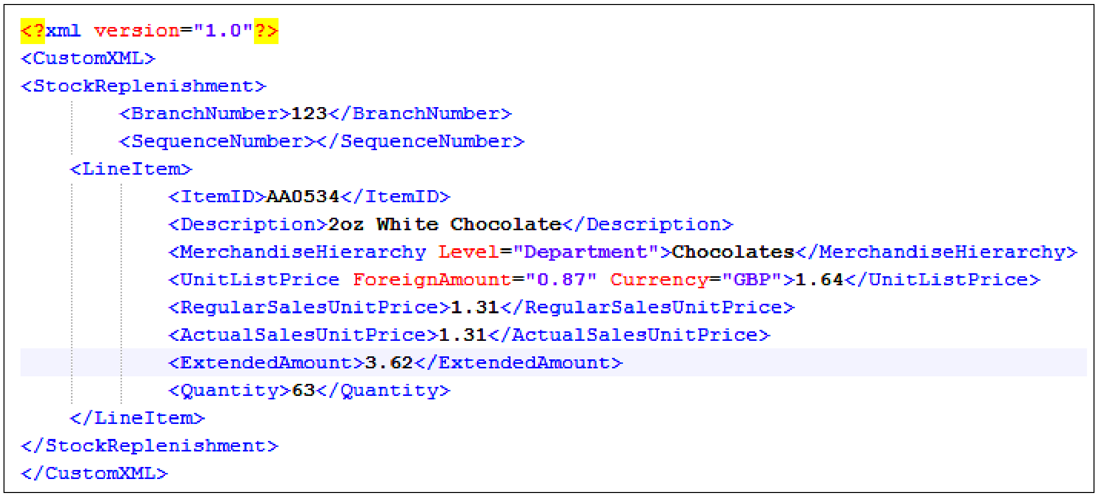
11.Figure 8-65 shows the file after it has been processed by the compute node. Notice that the routing information is removed and the file is in the format that the application expects.



```
   (0x01000000:Folder):XMLNSC     = ( ['xmlnsc' : 0x1a65dc60]
     (0x01000400:NamespaceDecl):XmlDeclaration = (
       (0x03000100:Attribute):Version  = '1.0' (CHARACTER)
       (0x03000100:Attribute):Encoding = 'UTF-8' (CHARACTER)
     )
     (0x01000000:Folder       ):CustomXML      = (
       (0x01000000:Folder):StockReplenishment = (
         (0x03000000:PCDataField):BranchNumber    = '123' (CHARACTER)
         (0x01000000:Folder      ):SequenceNumber  =
         (0x01000000:Folder      ):LineItem        = (
           (0x03000000:PCDataField):ItemID                = 'AA0534' (CHARACTER)
           (0x03000000:PCDataField):Description           = '2oz White
Chocolate' (CHARACTER)
           (0x03000000:PCDataField):MerchandiseHierarchy  = 'Chocolates'
(CHARACTER)
           (
             (0x03000100:Attribute):Level = 'Department' (CHARACTER)
           )
           (0x03000000:PCDataField):UnitListPrice         = '1.64' (CHARACTER)
           (
             (0x03000100:Attribute):ForeignAmount = '0.87' (CHARACTER)
             (0x03000100:Attribute):Currency      = 'GBP' (CHARACTER)
           )
           (0x03000000:PCDataField):RegularSalesUnitPrice = '1.31' (CHARACTER)
           (0x03000000:PCDataField):ActualSalesUnitPrice  = '1.31' (CHARACTER)
           (0x03000000:PCDataField):ExtendedAmount        = '3.62' (CHARACTER)
           (0x03000000:PCDataField):Quantity              = '63' (CHARACTER)
         )
       )
     )
   )
)
```

Figure 8-65   Trace file produced by AftPrepStk trace node

Figure 8-66 provides a better view of the file using an XML editor.



```
<?xml version="1.0"?>
<CustomXML>
<StockReplenishment>
        <BranchNumber>123</BranchNumber>
        <SequenceNumber></SequenceNumber>
    <LineItem>
            <ItemID>AA0534</ItemID>
            <Description>2oz White Chocolate</Description>
            <MerchandiseHierarchy Level="Department">Chocolates</MerchandiseHierarchy>
            <UnitListPrice ForeignAmount="0.87" Currency="GBP">1.64</UnitListPrice>
            <RegularSalesUnitPrice>1.31</RegularSalesUnitPrice>
            <ActualSalesUnitPrice>1.31</ActualSalesUnitPrice>
            <ExtendedAmount>3.62</ExtendedAmount>
            <Quantity>63</Quantity>
    </LineItem>
</StockReplenishment>
</CustomXML>
```

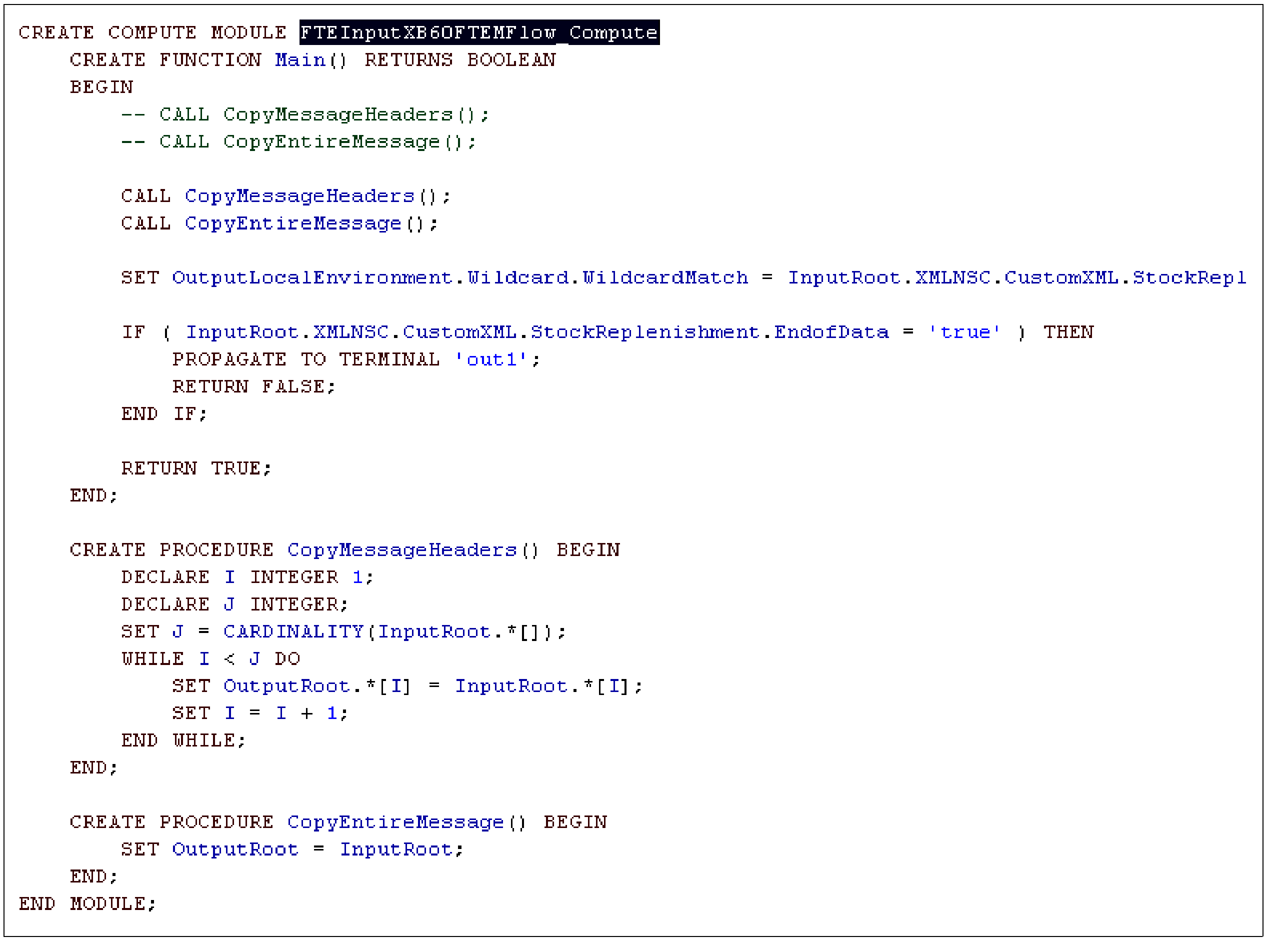
Figure 8-66 Message being sent to application after removal of route tags

12. Again referring to Figure 8-60 on page 241, you can see that the message is being routed to the application’s MQ queue via the “Sent to HQ Replenishment App (MQ)” MQOutput node. The name of the queue is XB60STKTOINV. By putting a message on the queue, the backend order inventory application receives the message and processes the order.

One of the strengths of WebSphere Message Broker is the routing of messages. As we mentioned earlier, the out terminal of the FTEInput node has two branches. The file is not only sent to the application for processing, but it is also being sent to the auditing department so that the original file sent from the branch can be archived for historical purposes (the upper branch in the flow shown in Figure 8-60 on page 241).

Assume that the archival system is running on another file server and receives files via the MQ FTE backbone. The FTEOutput node in the message flow sends the file to the AGTNFS destination agent on queue manager QMNFS. In our scenario, we use the same agent, but write to a different directory on the file system not shared with the XB60. In reality, we can use any agent in the MQ FTE backbone network.

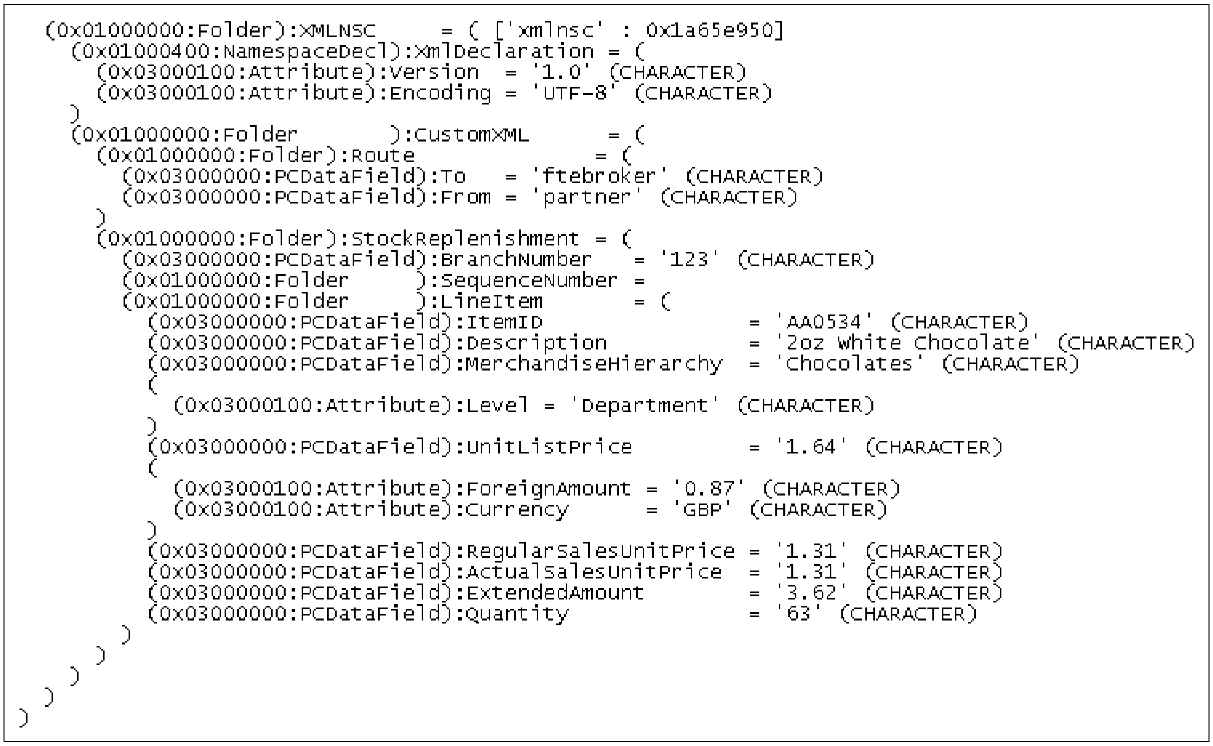
1. This flow is easy to do by adding an additional branch out of the FTEInput node. This path is routed through another compute node, PrepFTE. Figure 8-67 shows the ESQL code of the compute node.



```
CREATE COMPUTE MODULE FTEInputXB60FTEMFlow_Compute
    CREATE FUNCTION Main() RETURNS BOOLEAN
    BEGIN
        -- CALL CopyMessageHeaders();
        -- CALL CopyEntireMessage();

        CALL CopyMessageHeaders();
        CALL CopyEntireMessage();

        SET OutputLocalEnvironment.Wildcard.WildcardMatch = InputRoot.XMLNSC.CustomXML.StockRepl

        IF ( InputRoot.XMLNSC.CustomXML.StockReplenishment.EndofData = 'true' ) THEN
            PROPAGATE TO TERMINAL 'out1';
            RETURN FALSE;
        END IF;

        RETURN TRUE;
    END;

    CREATE PROCEDURE CopyMessageHeaders() BEGIN
        DECLARE I INTEGER 1;
        DECLARE J INTEGER;
        SET J = CARDINALITY(InputRoot.*[]);
        WHILE I < J DO
            SET OutputRoot.*[I] = InputRoot.*[I];
            SET I = I + 1;
        END WHILE;
    END;

    CREATE PROCEDURE CopyEntireMessage() BEGIN
        SET OutputRoot = InputRoot;
    END;
END MODULE;
```

Figure 8-67 ESQL code for sending and EOD data message to FTEOutput node

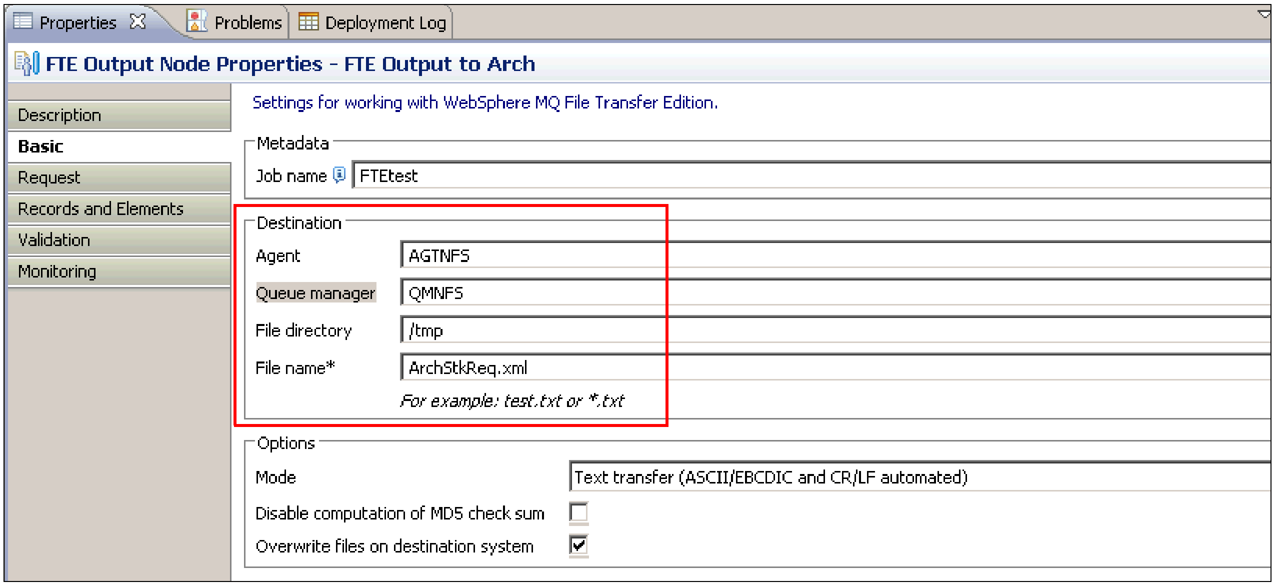
The data messages are routed to the out terminal. If we were parsing by record, and the test file had multiple records, then the compute node propagates an empty message to the FTEOutput signaling end of file. In this scenario, we parsed the whole file as a message, so only one message representing the whole file was sent to the AGTNFS agent from the BRKSAFE.DEFAULT agent to archive the file. Another trace node in the flow AftPrpArch records the message shown here.



```
  (0x01000000:Folder):XMLNSC     = ( ['xmlnsc' : 0x1a65e950]
    (0x01000400:NamespaceDecl):XmlDeclaration = (
      (0x03000100:Attribute):Version  = '1.0' (CHARACTER)
      (0x03000100:Attribute):Encoding = 'UTF-8' (CHARACTER)
    )
    (0x01000000:Folder       ):CustomXML      = (
      (0x01000000:Folder):Route             = (
        (0x03000000:PCDataField):To   = 'ftebroker' (CHARACTER)
        (0x03000000:PCDataField):From = 'partner' (CHARACTER)
      )
      (0x01000000:Folder):StockReplenishment = (
        (0x03000000:PCDataField):BranchNumber   = '123' (CHARACTER)
        (0x01000000:Folder     ):SequenceNumber =
        (0x01000000:Folder     ):LineItem       = (
          (0x03000000:PCDataField):ItemID                = 'AA0534' (CHARACTER)
          (0x03000000:PCDataField):Description           = '2oz White Chocolate' (CHARACTER)
          (0x03000000:PCDataField):MerchandiseHierarchy  = 'Chocolates' (CHARACTER)
          (
            (0x03000100:Attribute):Level = 'Department' (CHARACTER)
          )
          (0x03000000:PCDataField):UnitListPrice         = '1.64' (CHARACTER)
          (
            (0x03000100:Attribute):ForeignAmount = '0.87' (CHARACTER)
            (0x03000100:Attribute):Currency      = 'GBP' (CHARACTER)
          )
          (0x03000000:PCDataField):RegularSalesUnitPrice = '1.31' (CHARACTER)
          (0x03000000:PCDataField):ActualSalesUnitPrice  = '1.31' (CHARACTER)
          (0x03000000:PCDataField):ExtendedAmount        = '3.62' (CHARACTER)
          (0x03000000:PCDataField):Quantity              = '63' (CHARACTER)
        )
      )
    )
  )
)
```

Figure 8-68 Trace record of input file before sending to archival system via FTE

2. The FTEOutput node sends the file to AGTNFS. Figure 8-69 shows the properties.



FTE Output Node Properties - FTE Output to Arch

Settings for working with WebSphere MQ File Transfer Edition.

Metadata

Job name: FTEtest

Destination

Agent: AGTNFS

Queue manager: QMNFS

File directory: /tmp

File name*: ArchStkReq.xml

For example: test.txt or *.txt

Options

Mode: Text transfer (ASCII/EBCDIC and CR/LF automated)

Disable computation of MD5 check sum □

Overwrite files on destination system ☑

Figure 8-69 Properties for FTEOutput node to send file to archival system

Sending a XML document to external partner named partner (outbound flow)

The following flow explains how the tests were executed for the outbound flow.

1. As shown in Figure 8-70, the outbound message flow waits for an MQ message from the order inventory application. The flow is invoked when a message arrives on the queue specified in the properties of the MQInput node FromOrdInvApp. The queue name in the MQInput node is the same one that was defined for the MQOutput node in the inbound scenario.

   For testing this scenario, the outbound flow also simulates processing by a backend application. Consequently, the completion of the inbound flow initiates the outbound flow, which creates the response message to be sent back to the branch office.

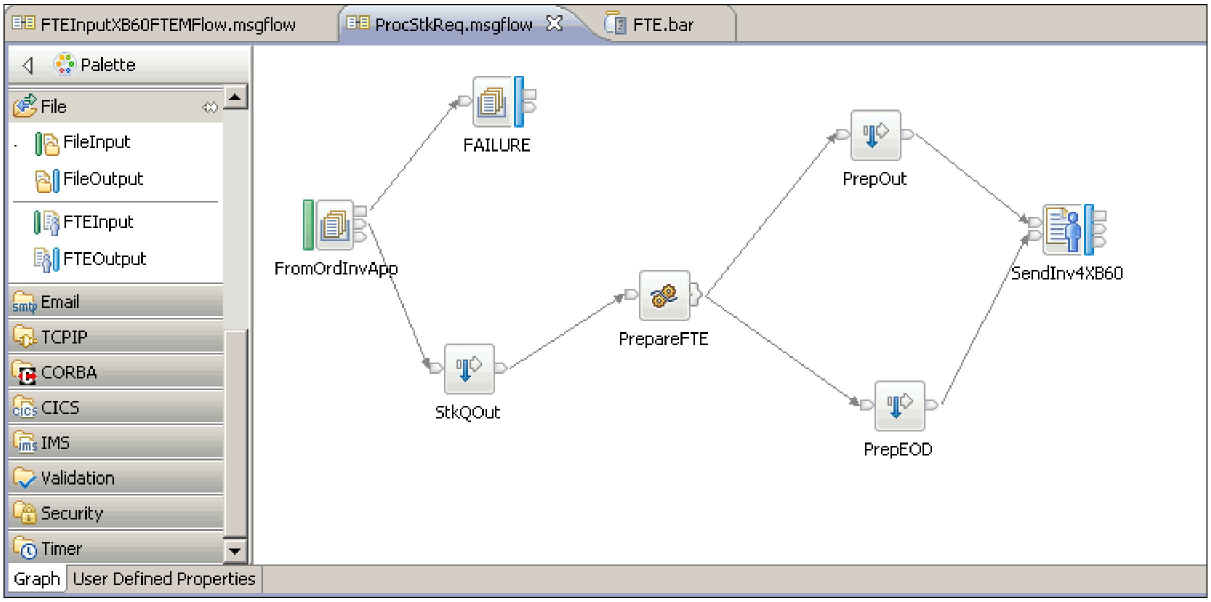
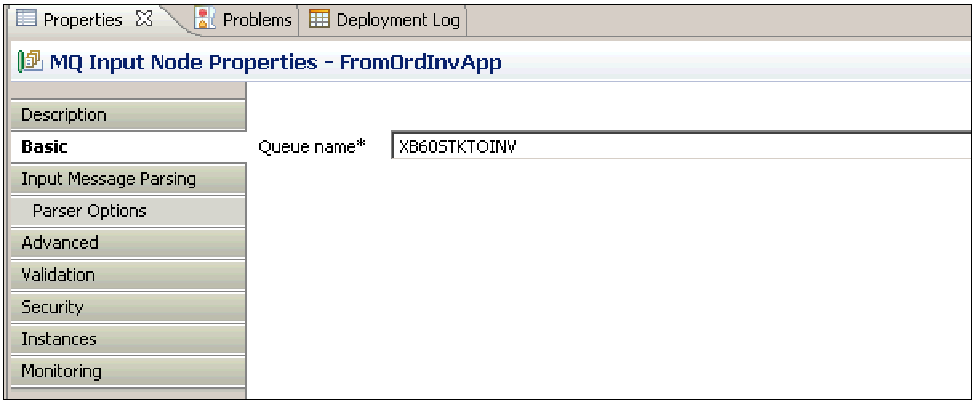


Figure 8-70 Outbound message flow to send response file from order - Inventory application

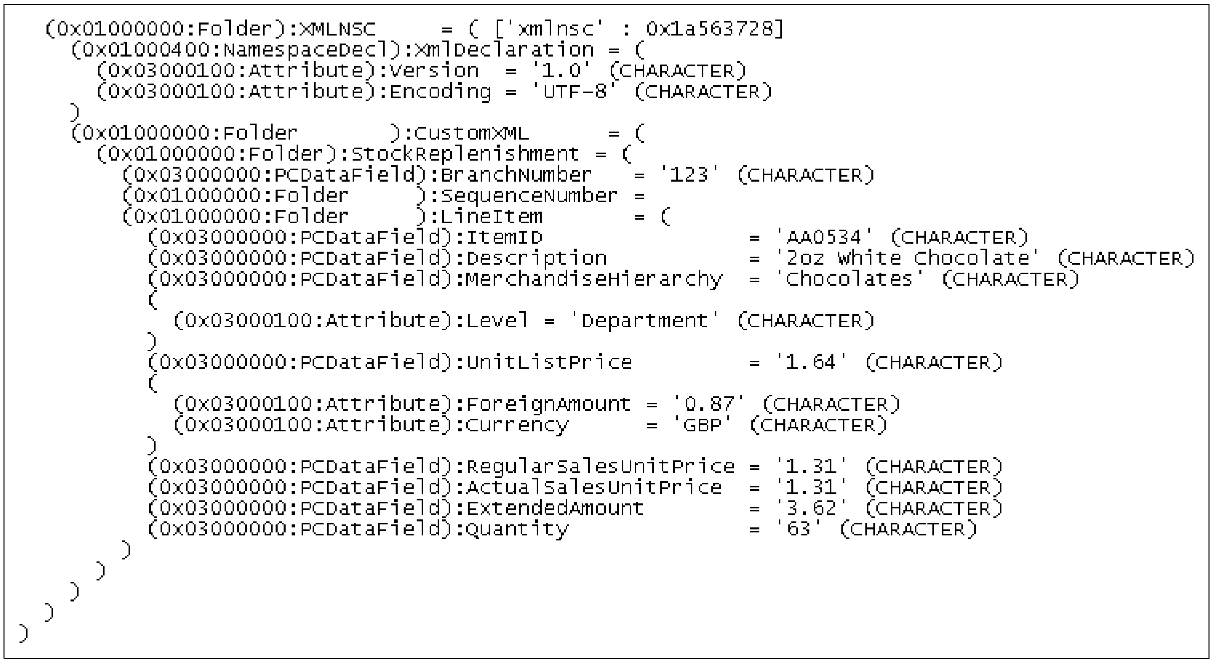
Figure 8-71 shows the properties of the MQInput node for the outbound flow.



Queue name* XB60STKTOINV

Figure 8-71 Outbound message flow - MQInput node FromOrdInvApp properties

2. Figure 8-72 is a display of the file as received from the application on the message flow's MQInput node from the XB60STKTOINV queue. The display was produced by the trace node StkQOut. You can see that it is the same file output by the inbound flow.



```
  (0x01000000:Folder):XMLNSC     = ( ['xmlnsc' : 0x1a563728]
    (0x01000400:NamespaceDecl):XmlDeclaration = (
      (0x03000100:Attribute):Version  = '1.0' (CHARACTER)
      (0x03000100:Attribute):Encoding = 'UTF-8' (CHARACTER)
    )
    (0x01000000:Folder       ):CustomXML      = (
      (0x01000000:Folder):StockReplenishment = (
        (0x03000000:PCDataField):BranchNumber   = '123' (CHARACTER)
        (0x01000000:Folder     ):SequenceNumber =
        (0x01000000:Folder     ):LineItem       = (
          (0x03000000:PCDataField):ItemID                 = 'AA0534' (CHARACTER)
          (0x03000000:PCDataField):Description            = '2oz White Chocolate' (CHARACTER)
          (0x03000000:PCDataField):MerchandiseHierarchy  = 'Chocolates' (CHARACTER)
          (
            (0x03000100:Attribute):Level = 'Department' (CHARACTER)
          )
          (0x03000000:PCDataField):UnitListPrice         = '1.64' (CHARACTER)
          (
            (0x03000100:Attribute):ForeignAmount = '0.87' (CHARACTER)
            (0x03000100:Attribute):Currency      = 'GBP' (CHARACTER)
          )
          (0x03000000:PCDataField):RegularSalesUnitPrice = '1.31' (CHARACTER)
          (0x03000000:PCDataField):ActualSalesUnitPrice  = '1.31' (CHARACTER)
          (0x03000000:PCDataField):ExtendedAmount        = '3.62' (CHARACTER)
          (0x03000000:PCDataField):Quantity              = '63' (CHARACTER)
        )
      )
    )
  )
)
```

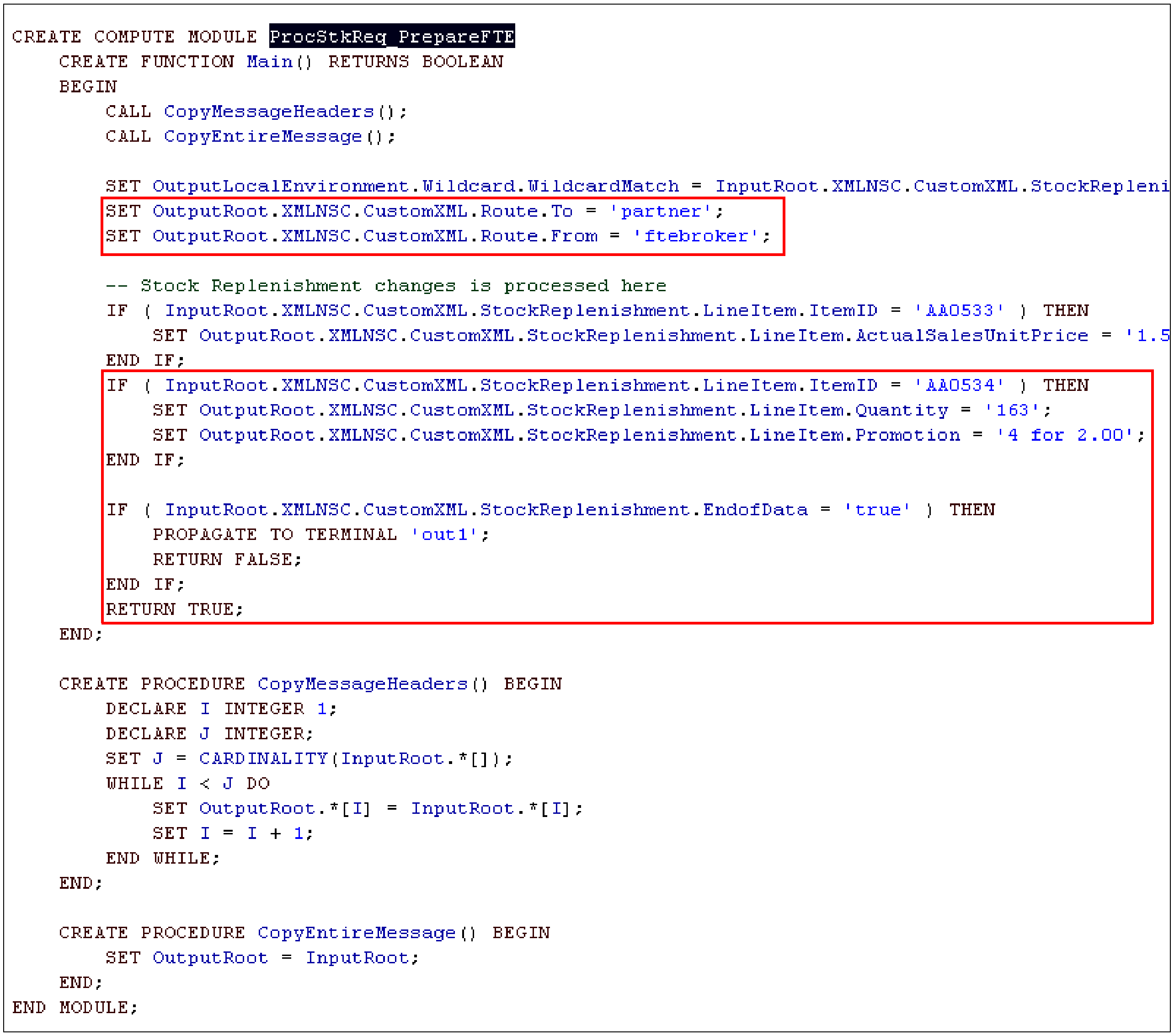
Figure 8-72 Trace node recording of message received on MQInput node

3. Before the file is sent to the XB60, the message flow must augment the file with routing information for the XB60. This is another strength of WebSphere Message Broker—transforming data. A compute node (PrepareFTE) has been inserted in the message follow to add the routing information.

   Note that, for the purposes of this scenario, the PrepareFTE compute node also simulates the back-end application's preparation of the response message. The response message contains updated stock availability and promotional information.

   We can override the destination properties for the FTEOutput node by using the LocalEnvironment.Destination.FTE variables, as discussed previously in "Local environment tree structure for FTE nodes" on page 230. However, in this scenario, we add routing information to the outbound file so that the XB60 can validate partners and send the file to the correct destination with the correct protocol. In this case, we use SFTP to send the file to external partner *partner's* SFTP file server.

Figure 8-73 shows the ESQL code from the PrepareFTE compute node. This is a simple ESQL example of adding data to the file. In this case, item quantities and promotional information is added.

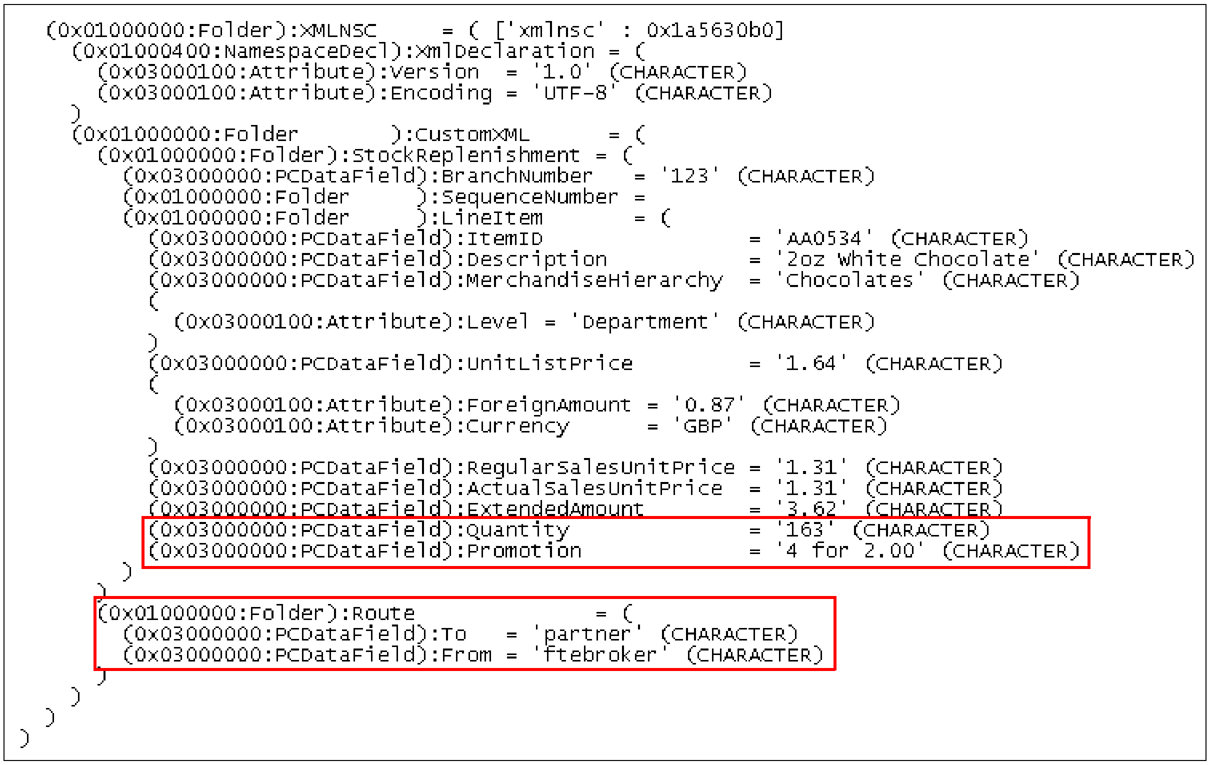


```
CREATE COMPUTE MODULE ProcStkReq_PrepareFTE
    CREATE FUNCTION Main() RETURNS BOOLEAN
    BEGIN
        CALL CopyMessageHeaders();
        CALL CopyEntireMessage();

        SET OutputLocalEnvironment.Wildcard.WildcardMatch = InputRoot.XMLNSC.CustomXML.StockRepleni
        SET OutputRoot.XMLNSC.CustomXML.Route.To = 'partner';
        SET OutputRoot.XMLNSC.CustomXML.Route.From = 'ftebroker';

        -- Stock Replenishment changes is processed here
        IF ( InputRoot.XMLNSC.CustomXML.StockReplenishment.LineItem.ItemID = 'AA0533' ) THEN
            SET OutputRoot.XMLNSC.CustomXML.StockReplenishment.LineItem.ActualSalesUnitPrice = '1.5
        END IF;
        IF ( InputRoot.XMLNSC.CustomXML.StockReplenishment.LineItem.ItemID = 'AA0534' ) THEN
            SET OutputRoot.XMLNSC.CustomXML.StockReplenishment.LineItem.Quantity = '163';
            SET OutputRoot.XMLNSC.CustomXML.StockReplenishment.LineItem.Promotion = '4 for 2.00';
        END IF;

        IF ( InputRoot.XMLNSC.CustomXML.StockReplenishment.EndofData = 'true' ) THEN
            PROPAGATE TO TERMINAL 'out1';
            RETURN FALSE;
        END IF;
        RETURN TRUE;
    END;

    CREATE PROCEDURE CopyMessageHeaders() BEGIN
        DECLARE I INTEGER 1;
        DECLARE J INTEGER;
        SET J = CARDINALITY(InputRoot.*[]);
        WHILE I < J DO
            SET OutputRoot.*[I] = InputRoot.*[I];
            SET I = I + 1;
        END WHILE;
    END;

    CREATE PROCEDURE CopyEntireMessage() BEGIN
        SET OutputRoot = InputRoot;
    END;
END MODULE;
```

Figure 8-73 ESQL code simulating application processing - Addition of routing information

Notice that the code also sets the destination and sender. The XB60 B2B gateway needs the routing information to validate the partners and extract the detailed routing addresses from the partner profile.

4. Figure 8-74 displays the file as it exits PrepareFTE compute node. The display was captured by trace node PrepOut.



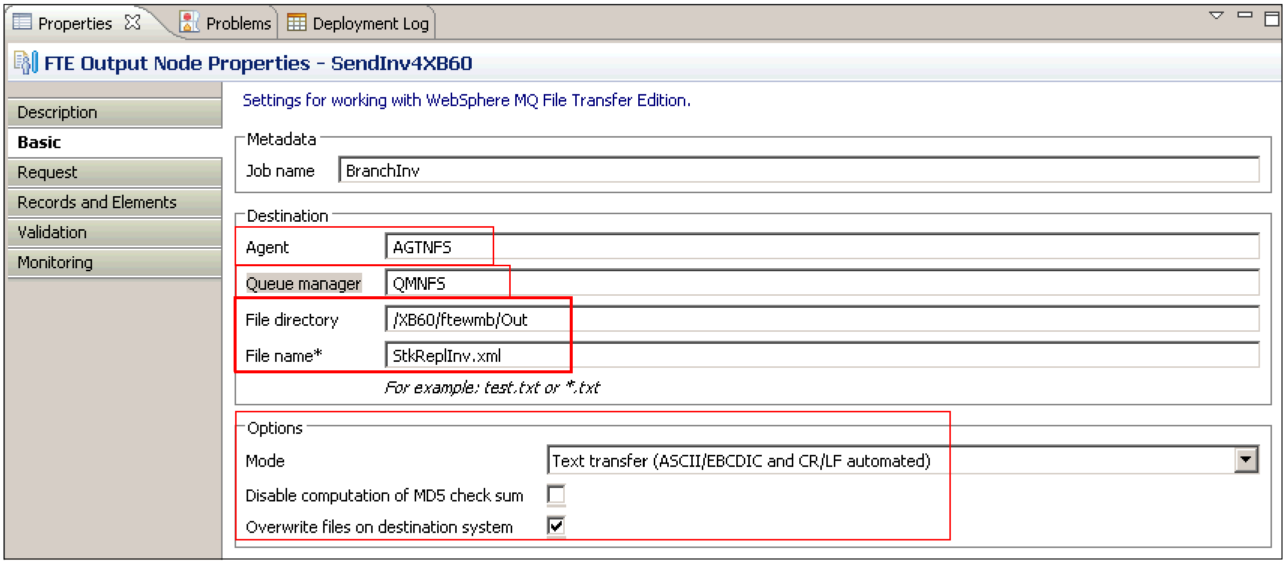
```
   (0x01000000:Folder):XMLNSC     = ( ['xmlnsc' : 0x1a5630b0]
     (0x01000400:NamespaceDecl):XmlDeclaration = (
       (0x03000100:Attribute):Version  = '1.0' (CHARACTER)
       (0x03000100:Attribute):Encoding = 'UTF-8' (CHARACTER)
     )
     (0x01000000:Folder       ):CustomXML      = (
       (0x01000000:Folder):StockReplenishment = (
         (0x03000000:PCDataField):BranchNumber   = '123' (CHARACTER)
         (0x01000000:Folder     ):SequenceNumber =
         (0x01000000:Folder     ):LineItem       = (
           (0x03000000:PCDataField):ItemID                  = 'AA0534' (CHARACTER)
           (0x03000000:PCDataField):Description             = '2oz White Chocolate' (CHARACTER)
           (0x03000000:PCDataField):MerchandiseHierarchy    = 'Chocolates' (CHARACTER)
           (
             (0x03000100:Attribute):Level = 'Department' (CHARACTER)
           )
           (0x03000000:PCDataField):UnitListPrice           = '1.64' (CHARACTER)
           (
             (0x03000100:Attribute):ForeignAmount = '0.87' (CHARACTER)
             (0x03000100:Attribute):Currency      = 'GBP' (CHARACTER)
           )
           (0x03000000:PCDataField):RegularSalesUnitPrice = '1.31' (CHARACTER)
           (0x03000000:PCDataField):ActualSalesUnitPrice  = '1.31' (CHARACTER)
           (0x03000000:PCDataField):ExtendedAmount        = '3.62' (CHARACTER)
           (0x03000000:PCDataField):Quantity              = '163' (CHARACTER)
           (0x03000000:PCDataField):Promotion             = '4 for 2.00' (CHARACTER)
         )
       )
       (0x01000000:Folder):Route              = (
         (0x03000000:PCDataField):To   = 'partner' (CHARACTER)
         (0x03000000:PCDataField):From = 'ftebroker' (CHARACTER)
       )
     )
   )
 )
```

Figure 8-74 Message after transformation - Routing tags added

Quantities have been changed and promotional information added. Even though the <*Route*></*Route*> tag has been added to the bottom of the data, xPath is used in the XB60 rules to locate the necessary information.

Now the message contains the data required by the B2B gateway and the FTEOutput node knows the agent and queue manager to which to send the file.

5. Referring to the FTEOutput node properties shown in Figure 8-75, we see that the file is to be sent via the MQ FTE backbone to the NFS file server, where the FTE_B2B_GW NFS poller is waiting for files. The FTEOutput node's properties are configured to instruct the broker's agent where to send the file.



FTE Output Node Properties - SendInv4XB60

Settings for working with WebSphere MQ File Transfer Edition.

Metadata

Job name: BranchInv

Destination

Agent: AGTNFS

Queue manager: QMNFS

File directory: /XB60/ftewmb/Out

File name*: StkReplInv.xml

*For example: test.txt or *.txt*

Options

Mode: Text transfer (ASCII/EBCDIC and CR/LF automated)

Disable computation of MD5 check sum ☐

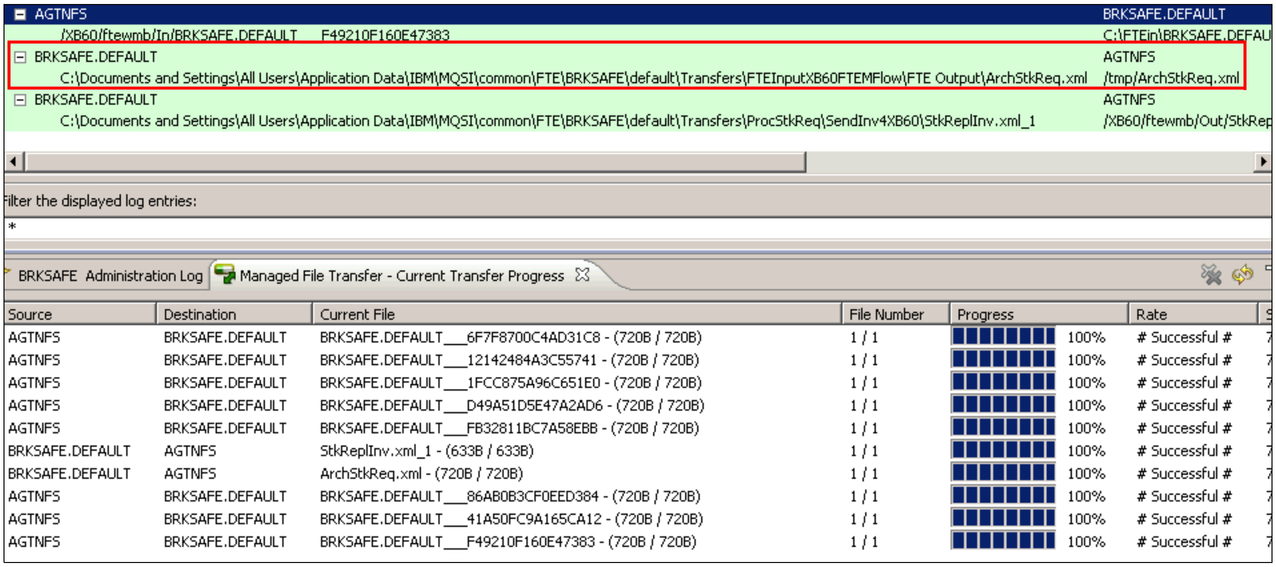
Overwrite files on destination system ☑

Figure 8-75 Outbound message flow FTEOuput node SendInv4XB60 properties

Looking at the properties shown in Figure 8-75, we can see the instructions for the broker's BRKSAFE.DEFAULT agent to send the file to the AGTNFS agent on queue manager QMNFS. The destination agent stores the `StkReplInv.xml` file in the `/xb60/ftewmb/Out` directory. The XB60 NFS poller is watching this directory for the arrival of files. Text mode and overwrite are specified as the options.

6. The file is sent to the AGTNFS agent from the broker's BRKSAFE.DEFAULT agent. AGTNFS runs on SYSE (the NFS file server) and writes the file in the `/xb60/ftewmb/Out` directory.

The FTEOutput node named the file `StkReplInv.xml`. We can verify this in the WebSphere MQ Explorer FTE Transfer Log (Figure 8-76). We can see the agents involved in the transfer and the path and file name where the file was stored by the destination agent.



```
⊟ AGTNFS                                                                                                                    BRKSAFE.DEFAULT
      /XB60/ftewmb/In/BRKSAFE.DEFAULT      F49210F160E47383                                                                 C:\FTEin\BRKSAFE.DEFAU
⊟ BRKSAFE.DEFAULT                                                                                                           AGTNFS
      C:\Documents and Settings\All Users\Application Data\IBM\MQSI\common\FTE\BRKSAFE\default\Transfers\FTEInputXB60FTEMFlow\FTE Output\ArchStkReq.xml   /tmp/ArchStkReq.xml
⊟ BRKSAFE.DEFAULT                                                                                                           AGTNFS
      C:\Documents and Settings\All Users\Application Data\IBM\MQSI\common\FTE\BRKSAFE\default\Transfers\ProcStkReq\SendInv4XB60\StkReplInv.xml_1   /XB60/ftewmb/Out/StkRep
```

Filter the displayed log entries:

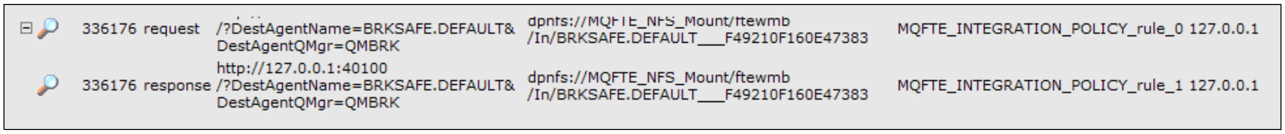
*

BRKSAFE Administration Log | Managed File Transfer - Current Transfer Progress

Source	Destination	Current File	File Number	Progress	Rate	S
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___6F7F8700C4AD31C8 - (720B / 720B)	1 / 1	100%	# Successful #	7
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___12142484A3C55741 - (720B / 720B)	1 / 1	100%	# Successful #	7
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___1FCC875A96C651E0 - (720B / 720B)	1 / 1	100%	# Successful #	7
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___D49A51D5E47A2AD6 - (720B / 720B)	1 / 1	100%	# Successful #	7
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___FB32811BC7A58EBB - (720B / 720B)	1 / 1	100%	# Successful #	7
BRKSAFE.DEFAULT	AGTNFS	StkReplInv.xml_1 - (633B / 633B)	1 / 1	100%	# Successful #	7
BRKSAFE.DEFAULT	AGTNFS	ArchStkReq.xml - (720B / 720B)	1 / 1	100%	# Successful #	7
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___86AB0B3CF0EED384 - (720B / 720B)	1 / 1	100%	# Successful #	7
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___41A50FC9A165CA12 - (720B / 720B)	1 / 1	100%	# Successful #	7
AGTNFS	BRKSAFE.DEFAULT	BRKSAFE.DEFAULT___F49210F160E47383 - (720B / 720B)	1 / 1	100%	# Successful #	7

Figure 8-76 WebSphere MQ Explorer FTE Transfer log

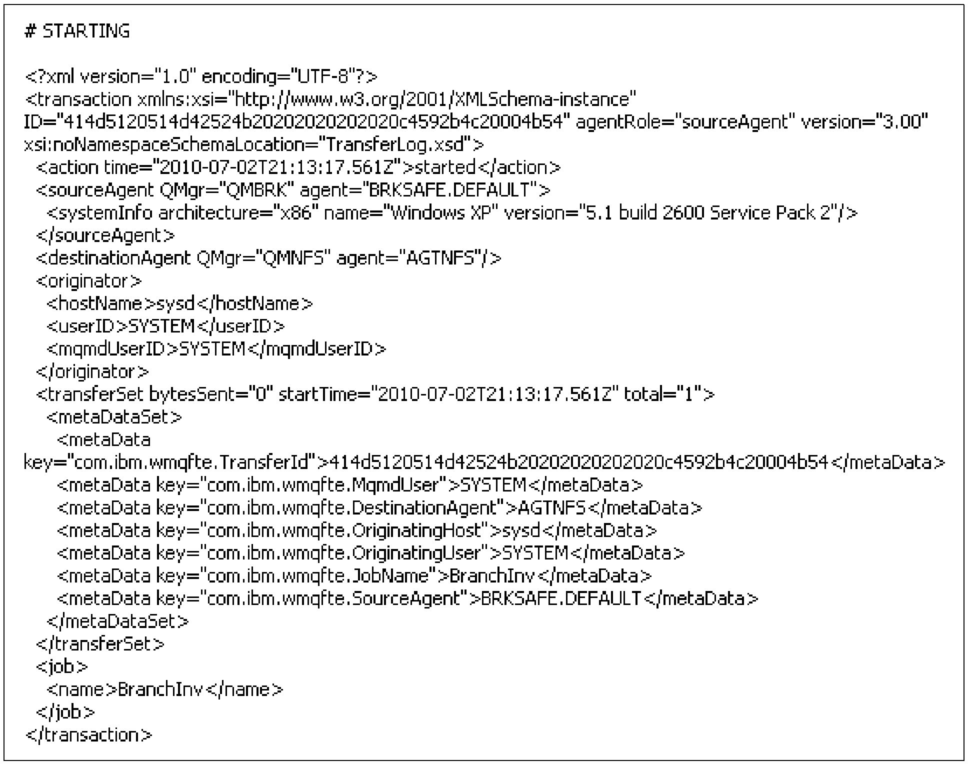
Figure 8-77 shows the corresponding entries found in the log of the XB60 NFS front side handler.



```
⊟ 336176 request  /?DestAgentName=BRKSAFE.DEFAULT&  dpnfs://MQFTE_NFS_Mount/ftewmb                 MQFTE_INTEGRATION_POLICY_rule_0 127.0.0.1
                  DestAgentQMgr=QMBRK               /In/BRKSAFE.DEFAULT___F49210F160E47383
                  http://127.0.0.1:40100
  336176 response /?DestAgentName=BRKSAFE.DEFAULT&  dpnfs://MQFTE_NFS_Mount/ftewmb                 MQFTE_INTEGRATION_POLICY_rule_1 127.0.0.1
                  DestAgentQMgr=QMBRK               /In/BRKSAFE.DEFAULT___F49210F160E47383
```

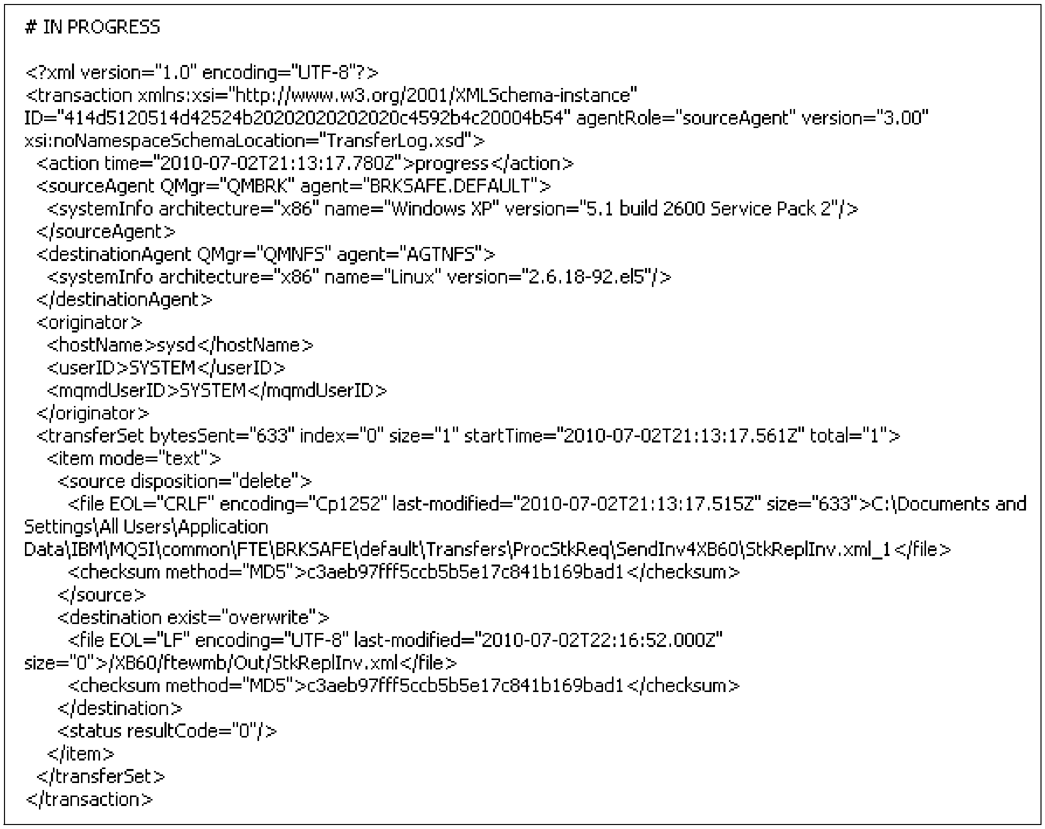
Figure 8-77 Corresponding entries in the XB60 log

7. In the WebSphere MQ File Transfer EditionTransfer Log, we right-click the transfer entry and select **Properties** to see the messages that were produced by the coordination queue manager relating to that particular transfer. Refer to Figure 8-78 through Figure 8-80 on page 256 to see the messages.



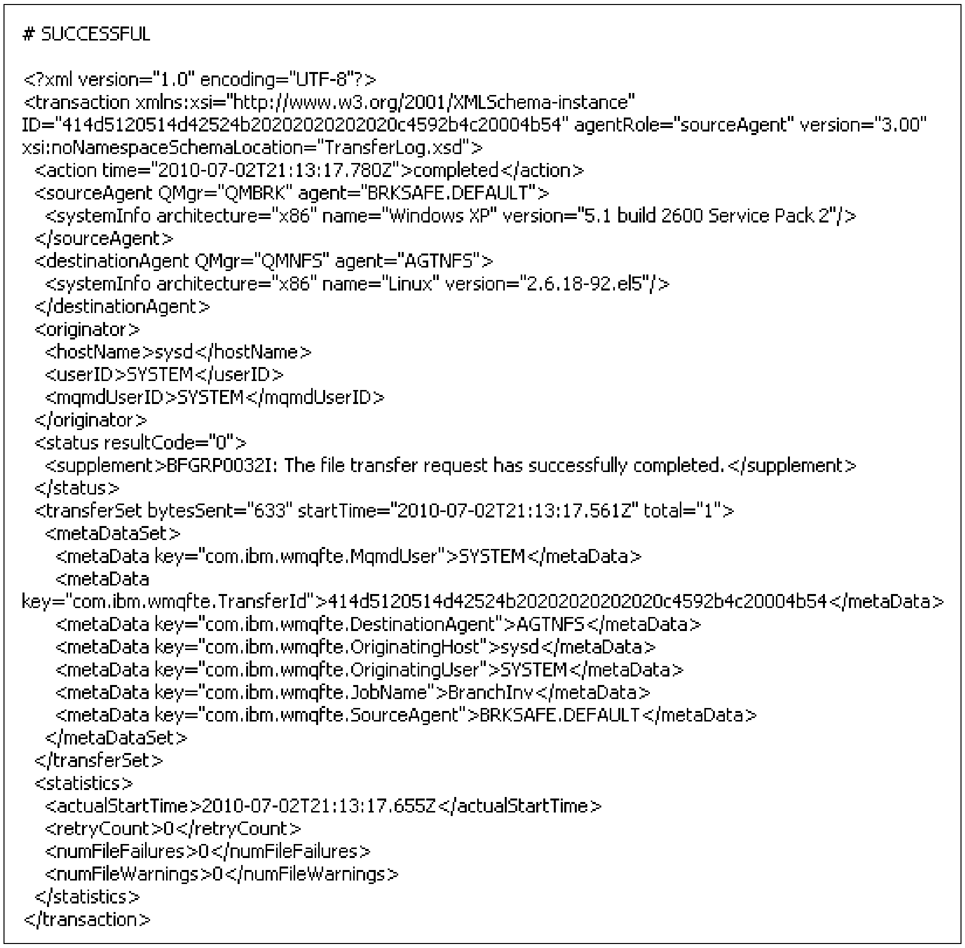
```
# STARTING

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
ID="414d5120514d42524b20202020202020c4592b4c20004b54" agentRole="sourceAgent" version="3.00"
xsi:noNamespaceSchemaLocation="TransferLog.xsd">
  <action time="2010-07-02T21:13:17.561Z">started</action>
  <sourceAgent QMgr="QMBRK" agent="BRKSAFE.DEFAULT">
    <systemInfo architecture="x86" name="Windows XP" version="5.1 build 2600 Service Pack 2"/>
  </sourceAgent>
  <destinationAgent QMgr="QMNFS" agent="AGTNFS"/>
  <originator>
    <hostName>sysd</hostName>
    <userID>SYSTEM</userID>
    <mqmdUserID>SYSTEM</mqmdUserID>
  </originator>
  <transferSet bytesSent="0" startTime="2010-07-02T21:13:17.561Z" total="1">
    <metaDataSet>
      <metaData
key="com.ibm.wmqfte.TransferId">414d5120514d42524b20202020202020c4592b4c20004b54</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">SYSTEM</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">AGTNFS</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">sysd</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">SYSTEM</metaData>
      <metaData key="com.ibm.wmqfte.JobName">BranchInv</metaData>
      <metaData key="com.ibm.wmqfte.SourceAgent">BRKSAFE.DEFAULT</metaData>
    </metaDataSet>
  </transferSet>
  <job>
    <name>BranchInv</name>
  </job>
</transaction>
```

Figure 8-78 Starting transfer message published by coordination queue manager QMSAFE



```
# IN PROGRESS

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
ID="414d5120514d42524b202020202020202020c4592b4c20004b54" agentRole="sourceAgent" version="3.00"
xsi:noNamespaceSchemaLocation="TransferLog.xsd">
  <action time="2010-07-02T21:13:17.780Z">progress</action>
  <sourceAgent QMgr="QMBRK" agent="BRKSAFE.DEFAULT">
    <systemInfo architecture="x86" name="Windows XP" version="5.1 build 2600 Service Pack 2"/>
  </sourceAgent>
  <destinationAgent QMgr="QMNFS" agent="AGTNFS">
    <systemInfo architecture="x86" name="Linux" version="2.6.18-92.el5"/>
  </destinationAgent>
  <originator>
    <hostName>sysd</hostName>
    <userID>SYSTEM</userID>
    <mqmdUserID>SYSTEM</mqmdUserID>
  </originator>
  <transferSet bytesSent="633" index="0" size="1" startTime="2010-07-02T21:13:17.561Z" total="1">
    <item mode="text">
      <source disposition="delete">
        <file EOL="CRLF" encoding="Cp1252" last-modified="2010-07-02T21:13:17.515Z" size="633">C:\Documents and
Settings\All Users\Application
Data\IBM\MQSI\common\FTE\BRKSAFE\default\Transfers\ProcStkReq\SendInv4XB60\StkReplInv.xml_1</file>
        <checksum method="MD5">c3aeb97fff5ccb5b5e17c841b169bad1</checksum>
      </source>
      <destination exist="overwrite">
        <file EOL="LF" encoding="UTF-8" last-modified="2010-07-02T22:16:52.000Z"
size="0">/XB60/ftewmb/Out/StkReplInv.xml</file>
        <checksum method="MD5">c3aeb97fff5ccb5b5e17c841b169bad1</checksum>
      </destination>
      <status resultCode="0"/>
    </item>
  </transferSet>
</transaction>
```

Figure 8-79 In Progress message published by coordination queue manager QMSAFE

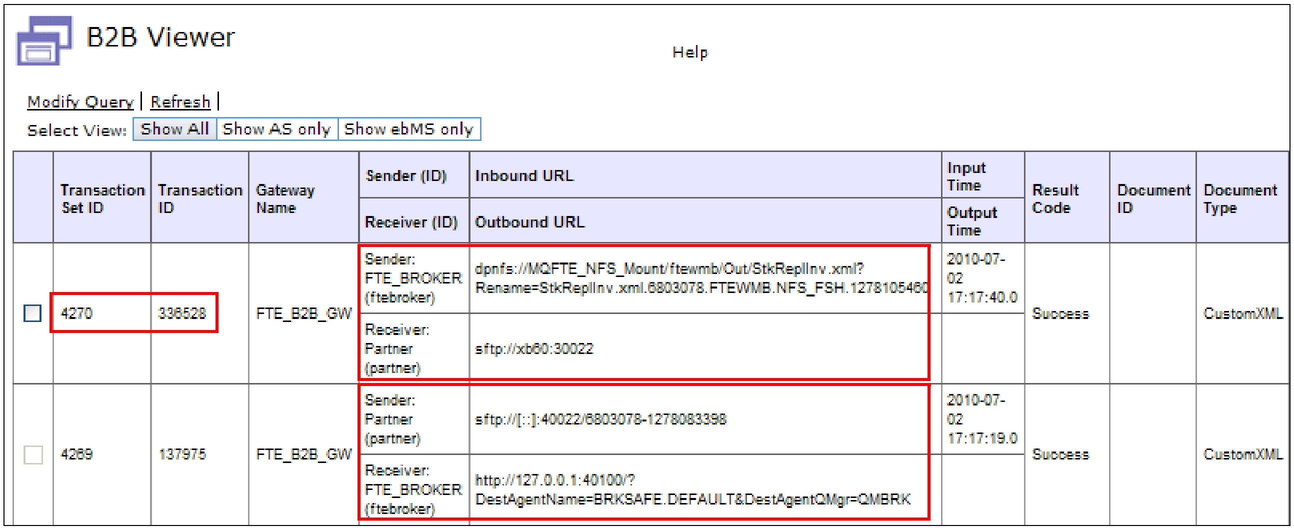


```
# SUCCESSFUL

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
ID="414d5120514d42524b20202020202020c4592b4c20004b54" agentRole="sourceAgent" version="3.00"
xsi:noNamespaceSchemaLocation="TransferLog.xsd">
  <action time="2010-07-02T21:13:17.780Z">completed</action>
  <sourceAgent QMgr="QMBRK" agent="BRKSAFE.DEFAULT">
    <systemInfo architecture="x86" name="Windows XP" version="5.1 build 2600 Service Pack 2"/>
  </sourceAgent>
  <destinationAgent QMgr="QMNFS" agent="AGTNFS">
    <systemInfo architecture="x86" name="Linux" version="2.6.18-92.el5"/>
  </destinationAgent>
  <originator>
    <hostName>sysd</hostName>
    <userID>SYSTEM</userID>
    <mqmdUserID>SYSTEM</mqmdUserID>
  </originator>
  <status resultCode="0">
    <supplement>BFGRP0032I: The file transfer request has successfully completed.</supplement>
  </status>
  <transferSet bytesSent="633" startTime="2010-07-02T21:13:17.561Z" total="1">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.MqmdUser">SYSTEM</metaData>
      <metaData
key="com.ibm.wmqfte.TransferId">414d5120514d42524b20202020202020c4592b4c20004b54</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">AGTNFS</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">sysd</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">SYSTEM</metaData>
      <metaData key="com.ibm.wmqfte.JobName">BranchInv</metaData>
      <metaData key="com.ibm.wmqfte.SourceAgent">BRKSAFE.DEFAULT</metaData>
    </metaDataSet>
  </transferSet>
  <statistics>
    <actualStartTime>2010-07-02T21:13:17.655Z</actualStartTime>
    <retryCount>0</retryCount>
    <numFileFailures>0</numFileFailures>
    <numFileWarnings>0</numFileWarnings>
  </statistics>
</transaction>
```

Figure 8-80 Successful message published by coordination queue manager QMSAFE

8. The routing information is simple. When the XB60 FTE_B2B_GW receives the file, all that it needs to know is whom the file is intended for and whom it is from. If the partners, FTE_BROKER and PARTNER, have valid profiles and there is a trading partner agreement in place for them, then the gateway has the information that it needs to complete the transfer.

Using the XB60 Transaction Viewer in the FTEWMB domain, we see in Transaction Set ID 4270, Transaction ID 336528 that the file was received from the FTE_BROKER partner and sent to partner via SFTP (Figure 8-81).



B2B Viewer

Help

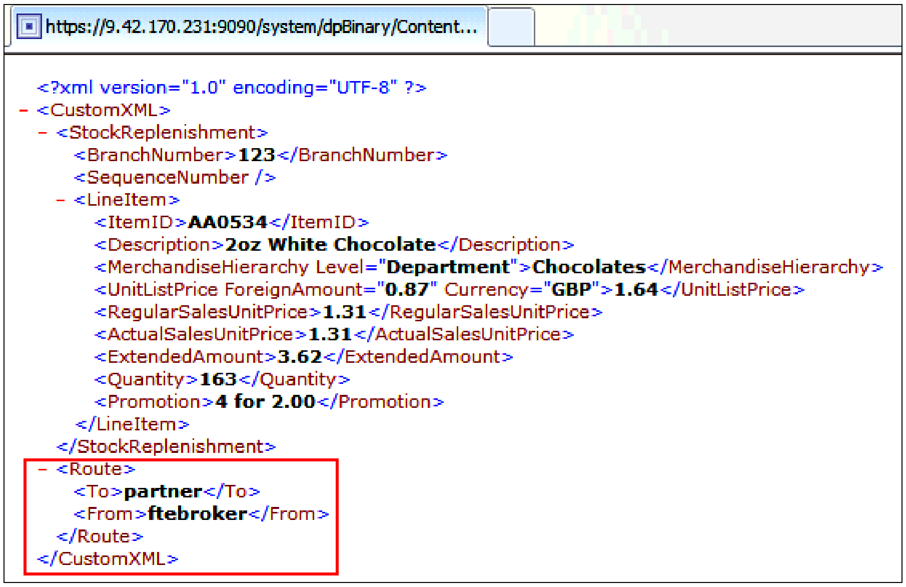
Modify Query | Refresh |

Select View: Show All | Show AS only | Show ebMS only

	Transaction Set ID	Transaction ID	Gateway Name	Sender (ID)	Inbound URL	Input Time	Result Code	Document ID	Document Type
				Receiver (ID)	Outbound URL	Output Time			
□	4270	336528	FTE_B2B_GW	Sender: FTE_BROKER (ftebroker)	dpnfs://MQFTE_NFS_Mount/ftewmb/Out/StkReplInv.xml?Rename=StkReplInv.xml.6803078.FTEWMB.NFS_FSH.1278105460	2010-07-02 17:17:40.0	Success		CustomXML
				Receiver: Partner (partner)	sftp://xb60:30022				
□	4269	137975	FTE_B2B_GW	Sender: Partner (partner)	sftp://[::]:40022/6803078-1278083398	2010-07-02 17:17:19.0	Success		CustomXML
				Receiver: FTE_BROKER (ftebroker)	http://127.0.0.1:40100/?DestAgentName=BRKSAFE.DEFAULT&DestAgentQMgr=QMBRK				

Figure 8-81 Outbound SFTP transaction from FTE_BROKER

Click the hot link for the Transaction Set ID 4270 and the data displays (Figure 8-82). This matches the data from the trace node (Figure 8-74 on page 251).

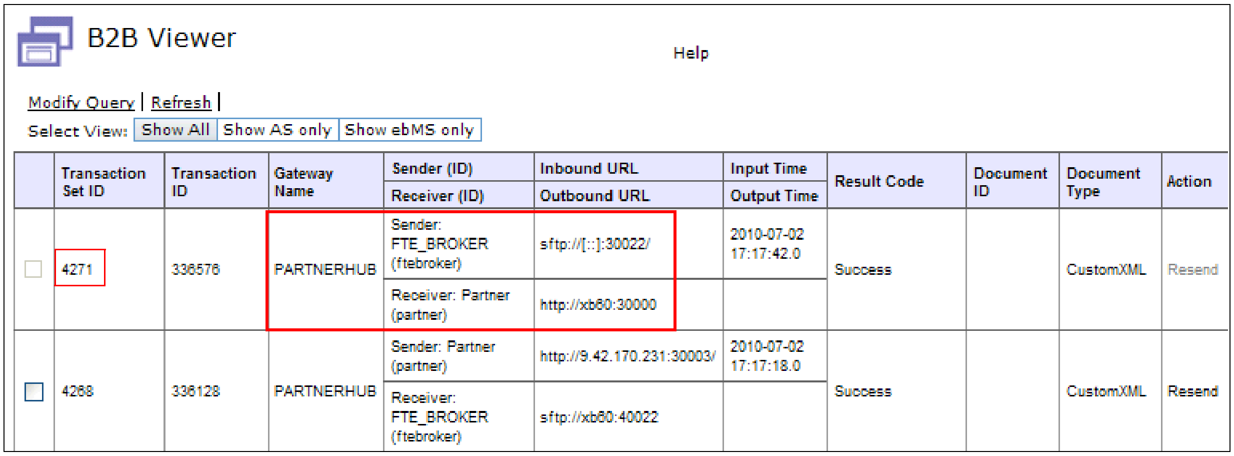


https://9.42.170.231:9090/system/dpBinary/Content...

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <CustomXML>
  - <StockReplenishment>
      <BranchNumber>123</BranchNumber>
      <SequenceNumber />
    - <LineItem>
        <ItemID>AA0534</ItemID>
        <Description>2oz White Chocolate</Description>
        <MerchandiseHierarchy Level="Department">Chocolates</MerchandiseHierarchy>
        <UnitListPrice ForeignAmount="0.87" Currency="GBP">1.64</UnitListPrice>
        <RegularSalesUnitPrice>1.31</RegularSalesUnitPrice>
        <ActualSalesUnitPrice>1.31</ActualSalesUnitPrice>
        <ExtendedAmount>3.62</ExtendedAmount>
        <Quantity>163</Quantity>
        <Promotion>4 for 2.00</Promotion>
      </LineItem>
    </StockReplenishment>
  - <Route>
      <To>partner</To>
      <From>ftebroker</From>
    </Route>
  </CustomXML>
```

Figure 8-82 Response file received by XB60 from NFS file system - Routing information included

9. The file is read by the NFS poller front side handler in the XB60's FTE_B2B_GW and sent via SFTP to the partner's SFTP front side handler in PARTNERHUB on port 30022. Looking at Figure 8-83, we see that the file was sent to the partner application, which was simulated by the NetTool request over HTTP.



B2B Viewer

Help

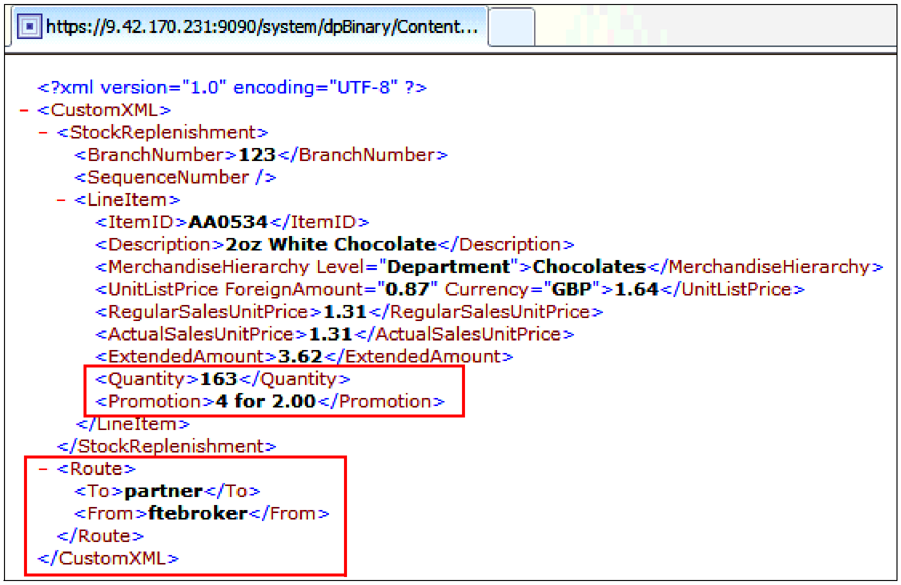
Modify Query | Refresh |

Select View: Show All | Show AS only | Show ebMS only

	Transaction Set ID	Transaction ID	Gateway Name	Sender (ID) Receiver (ID)	Inbound URL Outbound URL	Input Time Output Time	Result Code	Document ID	Document Type	Action
□	4271	336576	PARTNERHUB	Sender: FTE_BROKER (ftebroker) Receiver: Partner (partner)	sftp://[::]:30022/ http://xb60:30000	2010-07-02 17:17:42.0	Success		CustomXML	Resend
□	4268	336128	PARTNERHUB	Sender: Partner (partner) Receiver: FTE_BROKER (ftebroker)	http://9.42.170.231:30003/ sftp://xb60:40022	2010-07-02 17:17:18.0	Success		CustomXML	Resend

Figure 8-83 Outbound SFTP to NetTool HTTP

10. By right-clicking the hot link for Transaction Set ID 4271, we can view the content of the file going back to partner in Figure 8-84. We can see that this file exactly matches the file as read from the NFS in Figure 8-82 on page 257.



https://9.42.170.231:9090/system/dpBinary/Content...

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <CustomXML>
  - <StockReplenishment>
      <BranchNumber>123</BranchNumber>
      <SequenceNumber />
    - <LineItem>
        <ItemID>AA0534</ItemID>
        <Description>2oz White Chocolate</Description>
        <MerchandiseHierarchy Level="Department">Chocolates</MerchandiseHierarchy>
        <UnitListPrice ForeignAmount="0.87" Currency="GBP">1.64</UnitListPrice>
        <RegularSalesUnitPrice>1.31</RegularSalesUnitPrice>
        <ActualSalesUnitPrice>1.31</ActualSalesUnitPrice>
        <ExtendedAmount>3.62</ExtendedAmount>
        <Quantity>163</Quantity>
        <Promotion>4 for 2.00</Promotion>
      </LineItem>
    </StockReplenishment>
  - <Route>
      <To>partner</To>
      <From>ftebroker</From>
    </Route>
  </CustomXML>
```

Figure 8-84 Outbound response file sent to partner over SFTP

8.7 Troubleshooting tips for WebSphere Message Broker

In this section we provide troubleshooting information for WebSphere Message Broker and discuss how to diagnose errors that can occur in the scenarios described in this chapter. For more information, see:

- Troubleshooting for WebSphere MQ File Transfer Editionis discussed in 6.4, “Troubleshooting tips for WebSphere MQ File Transfer Edition” on page 105.
- Troubleshooting for the IBM WebSphere DataPower B2B Appliance XB60 is discussed in 7.6, “Troubleshooting tips” on page 170.
- Additional troubleshooting information for WebSphere Message Broker can be found at:

  http://publib.boulder.ibm.com/infocenter/wmbhelp/v7r0m0/topic/com.ibm.etools.mft.doc/bu03830_.htm
- Detailed information about WebSphere Message Broker diagnostic messages can be found at:

  http://publib.boulder.ibm.com/infocenter/wmbhelp/v7r0m0/topic/com.ibm.etools.mft.doc/ay66000_.htm

8.7.1 Check the system requirements

It is important for the installation, configuration, and operation of all the WebSphere Message Broker components that your systems are compliant with the system requirements for the software.

The system requirements can be found on the IBM web site at:

http://www-01.ibm.com/software/integration/wbimessagebroker/requirements/

If you are in doubt, contact IBM and ask for help.

8.7.2 Resource statistics

Resource statistics are collected by a broker to record performance and operating details of resources that are used by execution groups. The resource statistics can be used to ensure that your systems are using the available resources in the most efficient manner. If you detect that system resources are under pressure, you can examine the statistics collected by the broker to assess whether the cause of the concern is the use of those resources by processes in WebSphere Message Broker.

When resource statistics are activated in WebSphere Message Broker, measurements are collected for the resource type FTEAgent. Use these FTE agent resource manager statistics to view the number of transfers sent and received by an FTE agent. A one-to-one mapping exists between an execution group and its FTE agent. Consequently, the resource manager statistics provide an accurate picture of the file transfer activity of that execution group.

Table 8-7 describes the measurements that are returned for the agent. The measurements apply only to the 20-second interval that is being reported.

Table 8-7 FTE agent resource measurements

Measurements	Description
inboundTransfers	The number of transfers received by the agent
outboundTransfers	The number of transfers sent by the agent
inboundBytes	The number of bytes received by the agent
outboundBytes	The number of bytes sent by the agent

8.7.3 Log files

If an error is reported by a WebSphere Message Broker component, start investigation into its cause by looking at the product and systems logs to which information is written during component operation.

The information that is recorded in a log typically consists of a time stamp indicating when the error occurred and a brief description of the error, which can be expanded to provide more details. In certain situations, a general error is written and is followed by a group of more specific errors that provide details about the general error.

When an error occurs, check STDOUT, STDERR, and the local error log first. These logs record information about major activities within the system. All components of WebSphere Message Broker provide diagnostic information whenever error or warning conditions affect broker operation. These conditions include:

- Unsuccessful attempts to write a message to a WebSphere MQ output queue
- Errors interacting with databases
- The inability to parse an input message

Additional logs that are specific to WebSphere Message Broker are written to record runtime errors, internal errors that are produced by the operating system or your code, or errors related to the work that you are doing in a particular perspective, all of which you can view using the WebSphere Message Broker Toolkit.

There are a variety of logs that can be used to help with problem determination and troubleshooting. The location of the log files varies by platform. It is best to review the WebSphere Message Broker Information Center to find the location of the various log files on specific platforms. Information for interpreting the files can also be found at:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v7r0m0/topic/com.ibm.etools.mft.doc/an04120_.htm

8.7.4 User traces

If you cannot get enough information about a particular problem from the entries that are available in the various logs, the next troubleshooting method to consider is using trace. Trace provides more details about what is happening while code executes. The information produced from trace is sent to a specified trace record so that you or IBM support personnel can analyze it to discover the cause of your problem.

There are two main types of trace available in WebSphere Message Broker:

- User trace
- Service trace

Typically, you utilize user trace for debugging your applications. You can trace brokers, execution groups, and deployed message flows. With a service trace, you can activate more comprehensive broker tracing and start tracing for the WebSphere Message Broker Toolkit. You can also trace the execution of commands.

Trace is inactive by default, and must be explicitly activated by a command or by the WebSphere Message Broker Toolkit. When you start user tracing, you cause additional processing for every activity in the component that you are tracing. Large quantities of data are generated by the components. Expect performance to be affected while trace is active. You can limit this additional processing by being selective about what you trace and by restricting the time during which trace is active.

Before a broker or any of its execution groups or message flows can be traced, the broker must be running, and the message flows deployed.

To start a user trace:

1. Start WebSphere Message Broker user trace facilities by using the **`mqsichangetrace`** command, or, for execution groups and assigned message flows, from the WebSphere Message Broker Explorer. You can select only one broker on each invocation of the command, but you can activate concurrent traces for more than one broker by invoking the command more than once.
2. Specify an individual execution group or message flow within the specified broker to limit the scope of a trace. The events that are recorded when you select the message flow option include:
   - Sending a message from one message processing node to the next
   - Evaluating expressions in a filter or compute node
3. Start your trace. You can start trace at two levels:
   - normal: This tracks events that affect objects that you create and delete, such as nodes.
   - debug: This tracks the beginning and end of a process, and monitors objects that are affected by that process.

Example: Starting user trace for the default execution group

To start a normal level user trace for the default execution group on a broker that you have created with the name BRKSAFE, on distributed systems, enter the following command:

```
mqsichangetrace BRKSAFE -u -e default -l normal
```

Where:

- -u specifies user trace.
- -e specifies the execution group (in this case, the default execution group).
- -l specifies the level of trace (in this case, normal).

To stop the trace, enter:

```
mqsichangetrace BRKSAFE -u -e default -l none
```

Where:

- -u specifies user trace.
- -e specifies the execution group (in this case, the default execution group).
- -l specifies the level of trace (in this case, none).

On the z/OS platform, enter the following command:

```
F BRKSAFE,ct u=yes, e='default', l=normal
```

To stop the trace, enter:

```
F BRKSAFE,ct u=yes, e='default', l=none
```

To start normal level user tracing for the default execution group from the WebSphere Message Broker Explorer:

1. In the Navigator view, expand the **Brokers** folder and right-click the execution group with which you want to work.
2. Click **User Trace All Flows** → **Normal**.

To stop the trace:

1. In the Navigator view, expand the **Brokers** folder and right-click the execution group with which you want to work.
2. Click **User Trace All Flows** → **None**.

To start normal level user tracing for one of your message flows from the WebSphere Message Broker Explorer follow these instructions:

1. In the Navigator view, expand the **Brokers** folder and right-click the message flow with which you want to work.
2. Click **User Trace** → **Normal**.

To stop the trace:

1. In the Navigator view, expand the **Brokers** folder and right-click the message flow with which you want to work.
2. Click **User Trace** → **None**.

An alert saying that the message flow is tracing at the normal level displays in the Alert Viewer.

8.7.5 Common problems

This topic explains tactics for diagnosing problems when the message flows are run and the expected result is not received. Use the following instructions to diagnose the problem.

Use the WebSphere MQ Explorer to determine which queue the input message is on:

1. In WebSphere MQ Explorer, expand the folders to display the broker queue manager, QMBRK.
2. Click the **Queues** folder in the queue manager to display its queues.
3. Check the Current depth column to identify the queue that is holding the input message.

If several messages are stored on one queue, right-click the queue, then click **Browse Messages** to determine whether the message that you are interested in is on the queue.

One of the most common conditions experienced is that the message flow does start. Here are variations and what to do about them:

- Symptom: The input message stays on the MQInput queue.
  - The broker, the queue manager, the listener, or the message flow itself has stopped.

    Check that all the components are running and that the listener for the queue manager is listening on the port for the queue manager. Start any components that are not running.
  - Or, an unidentifiable message already on the IN queue cannot be processed by the message flow.

    In WebSphere MQ Explorer, right-click the **IN** queue, then click **All tasks** → **Clear Messages**.
- Symptom: An unidentifiable message already on the MQInput queue cannot be processed by the message flow.

  The MQInput node cannot identify which parser it must use to parse the message.

  If you are using the Enqueue facility in the workbench or the RfhUtil tool that is supplied in Support Pac IH03, you must type all the necessary message header information in the fields in the tool. If you are using the mqsiput.exe tool, you must add the header information to the message file itself.
- Symptom: The input message goes to the SYSTEM.DEAD.LETTER.QUEUE.

  The queue on which the input message was supposed to be put does not exist.

  Ensure that you have created all the queues required.
- Symptom: The input message goes to the FAIL queue.
  - The MQInput node cannot identify which parser it must use to parse the message.

    Check the Properties tab of the input node in the message flow. Make sure that the message domain matches the type of incoming message. For example, XMLNSC domain for XML message, IDOC for SAP, or MRM for COBOL copy books.

    Make sure that the incoming message format matches the message set defined for the input node of the message flow.
- Symptom: You cannot find the input message on any queue.
  - You have not refreshed the display in WebSphere MQ Explorer, or you have refreshed only some of the queues.

    To refresh all the queues in WebSphere MQ Explorer, right-click the **Queues** folder, then click **Refresh**. All the queues in the folder are refreshed.
  - Or, the input message was passed to a terminal that was not connected to another node, and the message was discarded.

    Ensure that all the nodes are connected to each other as required by the sample.

- Symptom: The message flow does not start when file arrives for a FTEInput node.
  - The execution group or message flow might be stopped.

    Use WebSphere Message Broker Explorer to start the message flow or the execution group.
  - If using temporary agents, the agent is not created until execution is started or the message flow with FTE nodes is deployed.

    In our testing, we noticed that files already in the input directory and not received by the FTE agent were not processed by the message flow.
  - Multiple flows might be deployed with FTEInput nodes using the same directory with matching file patterns.
    - Verify that the file was not processed by another flow.
    - Review properties for the FTEInput nodes and correct as needed.

8.7.6 Using the broker's queue manager as FTE coordination queue manager

When a message flow that contains an FTE node is deployed to an execution group, an agent is automatically created and started in that execution group. By default, the agent uses the broker's local queue manager as the coordination queue manager. If the broker's queue manager is being used as the coordination queue manager, the broker configures it as a coordination queue manager.

Unless you have previously defined the coordination queue manager, the agent is temporary. It is deleted when the flow is undeployed or the broker is stopped. This behavior is acceptable in a test environment. However, for production, the administrator must specify the coordination queue manager for the execution group.

Specifying a coordination queue manager:

- Ensures that the correct queue manager is used when the agent is created.
- Makes the agent permanent. If a coordination queue manager has been defined, the agent is deleted only after you undefine the coordination queue manager (for example, by setting it to an empty string) and restart the execution group.

A warning is written to the log if the coordination queue manager is not changed from the default.

This behavior caused confusion during the initial configuration of the solution scenarios. The state diagram (Figure 8-85) proved quite helpful and illustrates how the presence of nodes and a defined coordination queue manager affect the state of the agent.

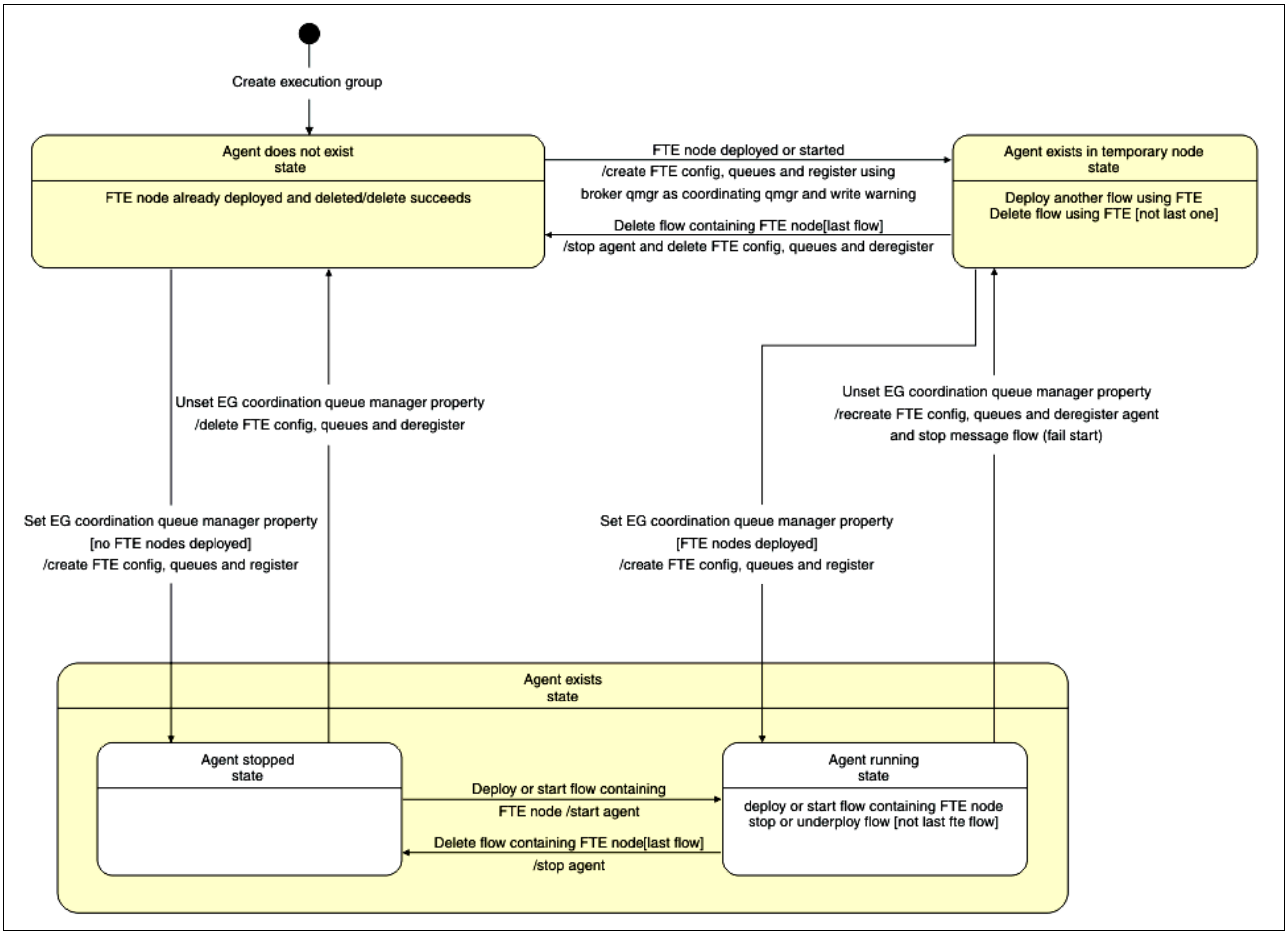

Create execution group
Agent does not exist state
FTE node already deployed and deleted/delete succeeds
FTE node deployed or started
/create FTE config, queues and register using
broker qmgr as coordinating qmgr and write warning
Delete flow containing FTE node[last flow]
/stop agent and delete FTE config, queues and deregister
Agent exists in temporary node state
Deploy another flow using FTE
Delete flow using FTE [not last one]
Unset EG coordination queue manager property
/delete FTE config, queues and deregister
Unset EG coordination queue manager property
/recreate FTE config, queues and deregister agent
and stop message flow (fail start)
Set EG coordination queue manager property
[no FTE nodes deployed]
/create FTE config, queues and register
Set EG coordination queue manager property
[FTE nodes deployed]
/create FTE config, queues and register
Agent exists state
Agent stopped state
Deploy or start flow containing
FTE node /start agent
Delete flow containing FTE node[last flow]
/stop agent
Agent running state
deploy or start flow containing FTE node
stop or underploy flow [not last fte flow]

Figure 8-85 How the presence of nodes and a defined coordination queue manager affect the state of the agents

The leading practice is to predefine the coordination queue manager and set the broker property to use that queue manager instead of the broker's queue manager.

Part 3

Appendixes

This part contains auxiliary information that you might find useful for understanding the scenarios. This part contains the following appendixes:

- Appendix A, “Configuration of WebSphere MQ File Transfer Edition” on page 269
- Appendix B, “Preparing the WebSphere Application Server and IBM HTTP Server environment” on page 297
- Appendix C, “Additional material” on page 317

A

Configuration of WebSphere MQ File Transfer Edition

This chapter describes how we configured WebSphere MQ File Transfer Edition for use with scenarios presented in Chapter 4, “Scenario topology overview” on page 37, through Chapter 7, “B2B-enabled managed file transfer” on page 109.

This chapter contains the following sections:

Overview

This chapter describes how we configured WebSphere MQ File Transfer Edition V7.0.2. Figure A-1 shows the topology we use.

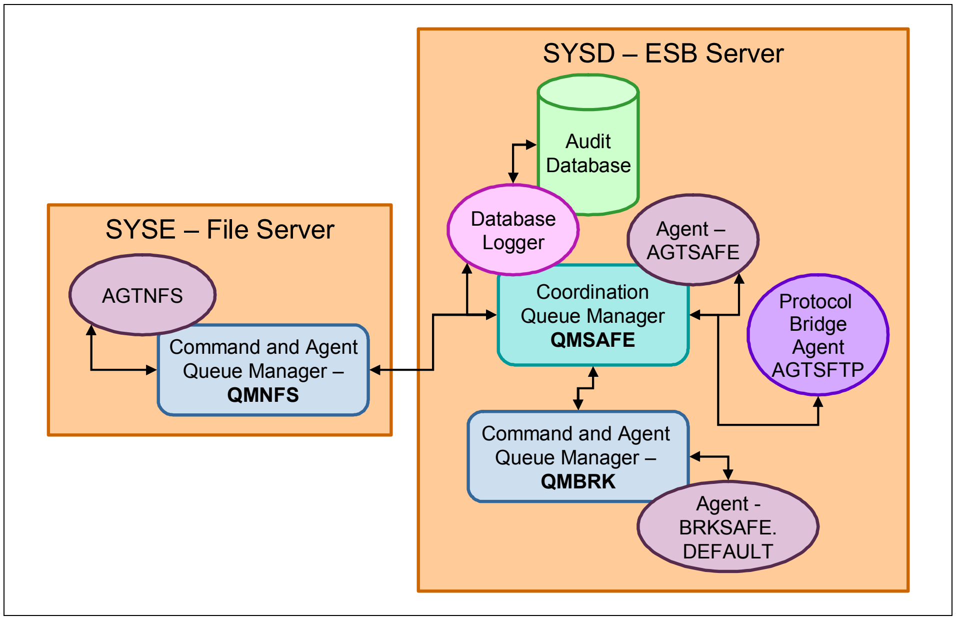

SYSD – ESB Server
Audit Database
Database Logger
Agent – AGTSAFE
Coordination Queue Manager QMSAFE
Protocol Bridge Agent AGTSFTP
Command and Agent Queue Manager – QMBRK
Agent - BRKSAFE. DEFAULT
SYSE – File Server
AGTNFS
Command and Agent Queue Manager – QMNFS

Figure A-1 Topology

In this chapter, we show how to set up all components shown in Figure A-1 except for AGTSFTP and BRKSAFE. The creation of these agents requires special treatment. AGTSFTP is described in 6.2.8, “Creating and configuring the bridge agent” on page 96. BRKSAFE.DEFAULT is described in “WebSphere MQ File Transfer Edition agents” on page 181.

Our configuration process proceeds as follows:

1. Create queue managers QMNFS, QMSAFE, and QMBRK.
2. Create an WebSphere MQ cluster that includes the three queue managers:
   - QMNFS
   - QMSAFE
   - QMBRK
3. Define queue manager QMSAFE as both the coordination and command queue manager for our WebSphere MQ File Transfer Edition network.
4. Create WebSphere MQ File Transfer Edition Server agents on QMNFS and QMSAFE.
   - AGTNFS on QMNFS
   - AGTSAFE on QMSAFE
5. Create the database logger.

Configuring WebSphere MQ

In this section, we create three queue managers:

- QMNFS
- QMSAFE
- QMBRK

Then we configure them as a cluster.

Creating the queue managers

Figure A-2 shows the starting point for our configuration process.

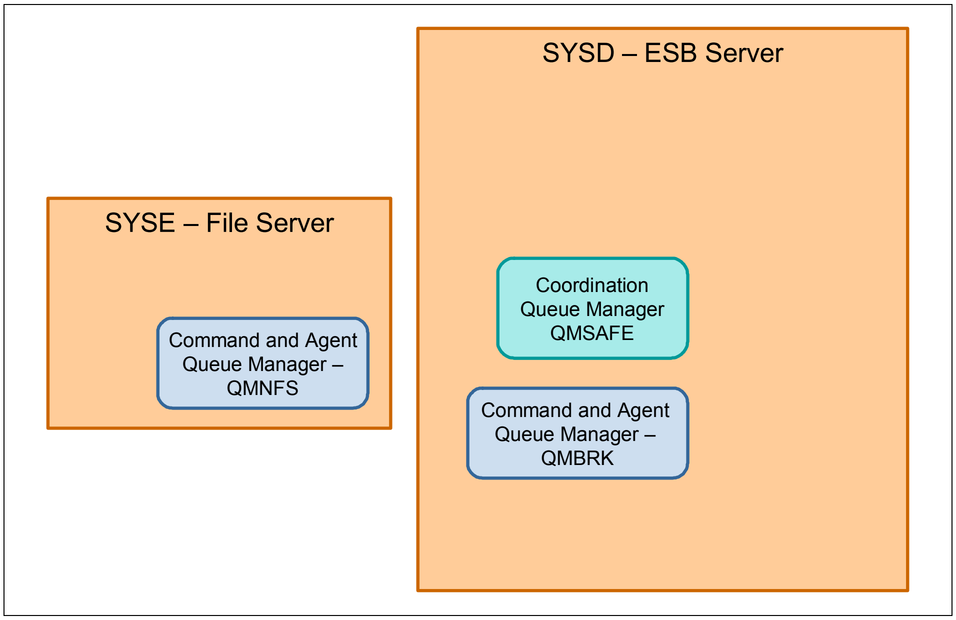

SYSD – ESB Server
SYSE – File Server
Command and Agent Queue Manager – QMNFS
Coordination Queue Manager QMSAFE
Command and Agent Queue Manager – QMBRK

Figure A-2 Queue managers for configuration of scenarios

Table A-1 summarizes the queue managers' topology.

Table A-1 Queue manager topology

Queue manager name	System	Port
QMSAFE	sysd	14014
QMBRK	sysd	14015
QMNFS	syse	1415

The queue managers can be created using the WebSphere MQ Explorer (on Windows and on Linux) or using the command-line interface (CLI). We created the queue managers using the command-line commands.

Creating queue managers QMSAFE and QMBRK on sysd

Example A-1 shows the commands that create the queue manager QMSAFE and the listener on port 14014, and define that the listener starts at queue manager startup.

Example A-1 Create queue manager QMSAFE

```
crtmqm -u QMSAFE.DLQ QMSAFE

strmqm QMSAFE
echo DEF LISTENER(LISTENER.TCP) TRPTYPE(TCP) PORT(14014) CONTROL(QMGR) | runmqsc
QMSAFE
echo START LISTENER(LISTENER.TCP) | runmqsc QMSAFE
```

Example A-2 shows the commands that create the queue manager QMBRK and the listener on port 14015, and define that the listener starts at queue manager startup.

Example A-2 Create queue manager QMBRK

```
crtmqm -u QMBRK.DLQ QMBRK

strmqm QMBRK
echo DEF LISTENER(LISTENER.TCP) TRPTYPE(TCP) PORT(14015) CONTROL(QMGR) | runmqsc
QMBRK
echo START LISTENER(LISTENER.TCP) | runmqsc QMBRK
```

Creating queue manager QMNFS on syse

Example A-3 shows the commands that create the queue manager QMNFS and the listener on port 1415, and define that the listener starts at queue manager startup.

Example A-3 Create queue manager QMNFS

```
crtmqm -u QMNFS.DLQ QMNFS

strmqm QMNFS
echo DEF LISTENER(LISTENER.TCP) TRPTYPE(TCP) PORT(1415) CONTROL(QMGR) | runmqsc
QMNFS
echo START LISTENER(LISTENER.TCP) | runmqsc QMNFS
```

Creating the queue manager cluster

The foundation of our file transfer scenarios is a WebSphere MQ messaging infrastructure. As can be seen from Figure A-3, we chose to configure our three queue managers as a cluster. This architecture has the benefits of simplicity, flexibility, ease of administration, and, in a production environment, load balancing. This section discusses the creation of the WebSphere MQ cluster configuration.

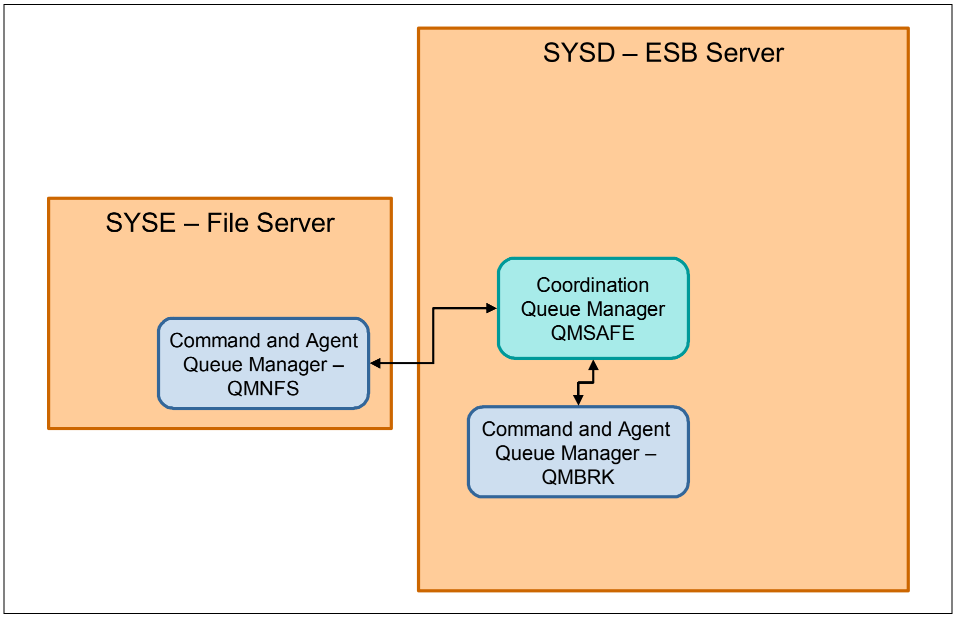

SYSD – ESB Server
SYSE – File Server
Command and Agent Queue Manager – QMNFS
Coordination Queue Manager QMSAFE
Command and Agent Queue Manager – QMBRK

Figure A-3 WebSphere MQ cluster configured

For more detailed information about how to set up an MQ v7 cluster see:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.csqzah.doc/qc10120_.htm

For our cluster we made the following decisions:

- The queue managers on sysd (QMSAFE and QMBRK) are full repository queue managers.
- The queue manager on syse (QMNFS) is a partial repository queue manager.

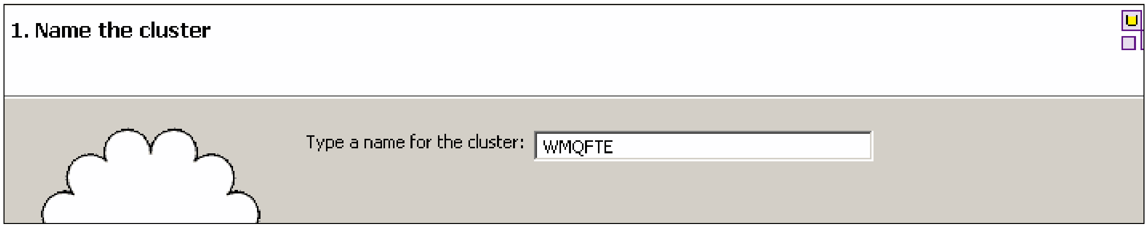
A cluster configuration can be created using either the WebSphere MQ Explorer GUI or using MQSC command-line scripts. We use the WebSphere MQ Explorer to create the cluster, because this tool provides a single point for defining the cluster and adding local and remote queue managers to it. Especially for users who have never defined a queue manager cluster, the WebSphere MQ Explorer makes it easy to go through all the steps of creating the cluster and the required channels.

In addition to the use of WebSphere MQ Explorer, we also show an example of a script for the creation of a queue manager cluster.

Creating the queue manager cluster

To create the queue manager cluster:

1. In the WebSphere MQ Explorer on the left pane, right-click **Queue Manager Clusters** and then click **New** → **Queue manager Cluster**. Enter the name of the cluster (Figure A-4).



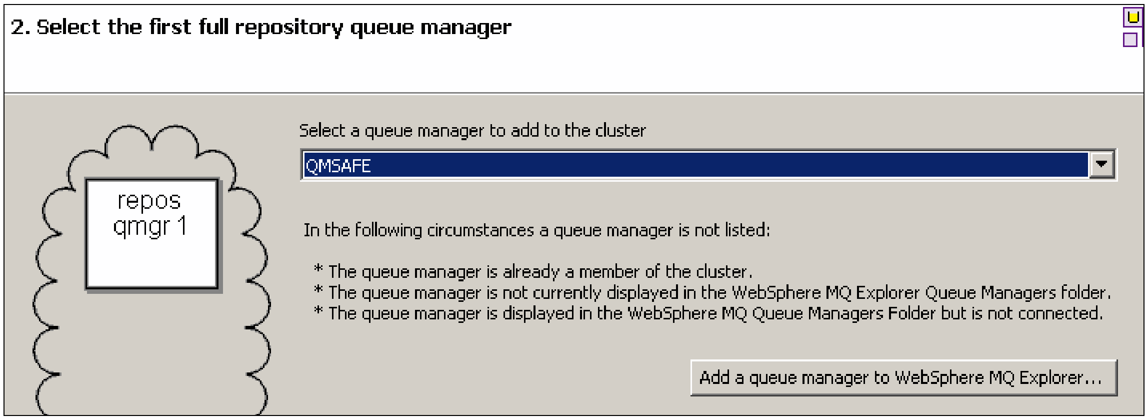
1. Name the cluster

Type a name for the cluster: WMQFTE

Figure A-4 Enter cluster name

   Click **Next**.

2. On the Select the first full repository queue manager panel, select the first full repository queue manager from the drop-down menu (Figure A-5).



2. Select the first full repository queue manager

Select a queue manager to add to the cluster: QMSAFE

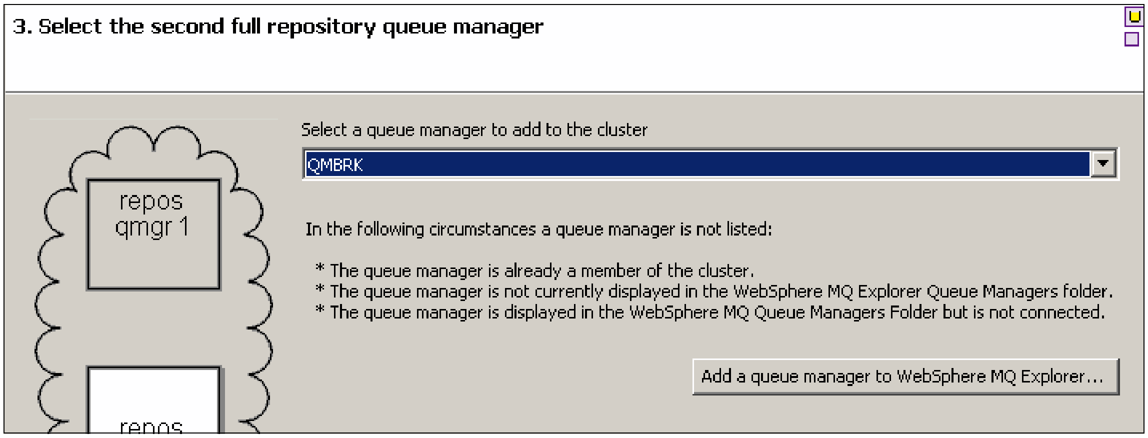
In the following circumstances a queue manager is not listed:

* The queue manager is already a member of the cluster.
* The queue manager is not currently displayed in the WebSphere MQ Explorer Queue Managers folder.
* The queue manager is displayed in the WebSphere MQ Queue Managers Folder but is not connected.

Figure A-5 Define first full repository queue manager

   Click **Next**.

3. On the Select the second full repository queue manager panel, select the second full repository queue manager from the drop-down menu (Figure A-6).



3. Select the second full repository queue manager

Select a queue manager to add to the cluster: QMBRK

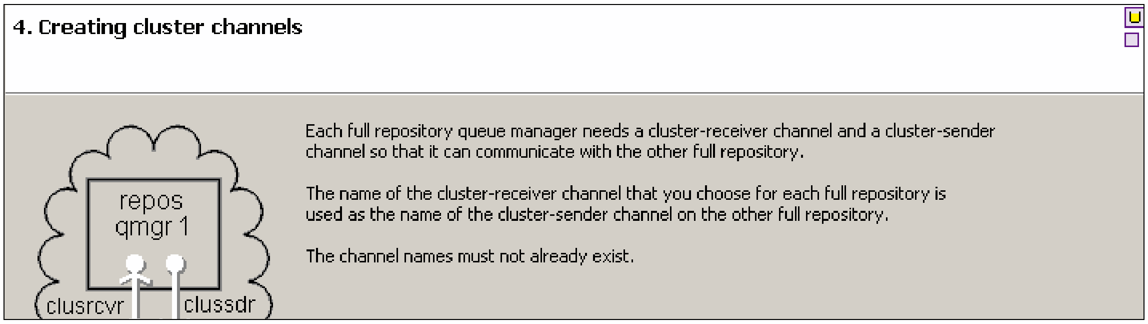
In the following circumstances a queue manager is not listed:

* The queue manager is already a member of the cluster.
* The queue manager is not currently displayed in the WebSphere MQ Explorer Queue Managers folder.
* The queue manager is displayed in the WebSphere MQ Queue Managers Folder but is not connected.

Figure A-6 Define second full repository queue manager

   Click **Next**.

On the Creating cluster channels panel, the channels between the full repository queue managers are defined (Figure A-7).



4. Creating cluster channels

Each full repository queue manager needs a cluster-receiver channel and a cluster-sender channel so that it can communicate with the other full repository.

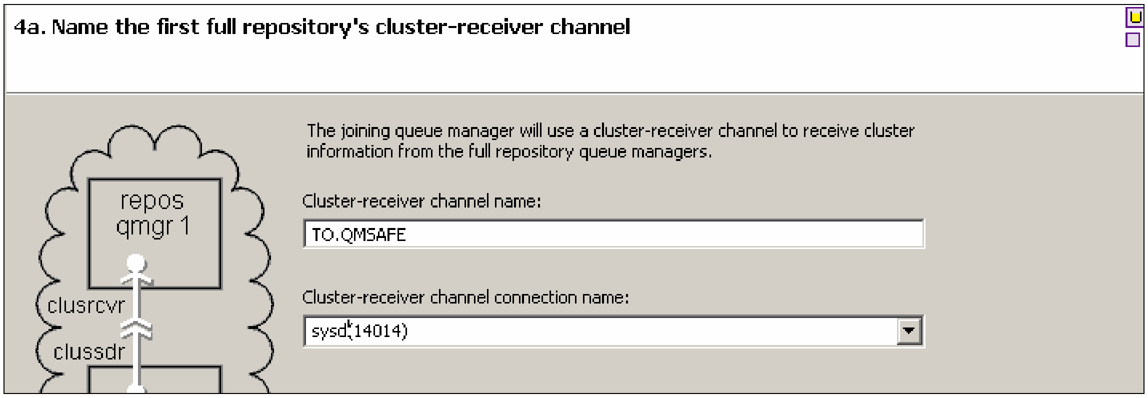
The name of the cluster-receiver channel that you choose for each full repository is used as the name of the cluster-sender channel on the other full repository.

The channel names must not already exist.

Figure A-7 Define channels between full repository queue managers

Click **Next**.

4. On the Name the first full respository's cluster receiver channel panel, for the first full repository cluster-receiver channel, enter the host name or IP address for the connection name (Figure A-8).



4a. Name the first full repository's cluster-receiver channel

The joining queue manager will use a cluster-receiver channel to receive cluster information from the full repository queue managers.

Cluster-receiver channel name:

TO.QMSAFE

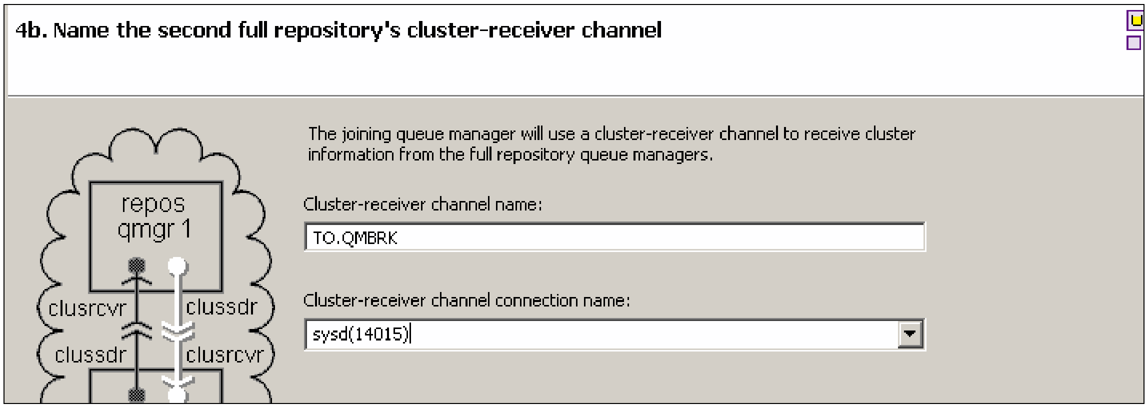
Cluster-receiver channel connection name:

sysd(14014)

Figure A-8 Details of first full repository receiver channel

Click **Next**.

5. On the Name the second full respository's cluster receiver channel panel, for the second full repository cluster-receiver channel, enter the host name or IP address for the connection name (Figure A-9).



4b. Name the second full repository's cluster-receiver channel

The joining queue manager will use a cluster-receiver channel to receive cluster information from the full repository queue managers.

Cluster-receiver channel name:

TO.QMBRK

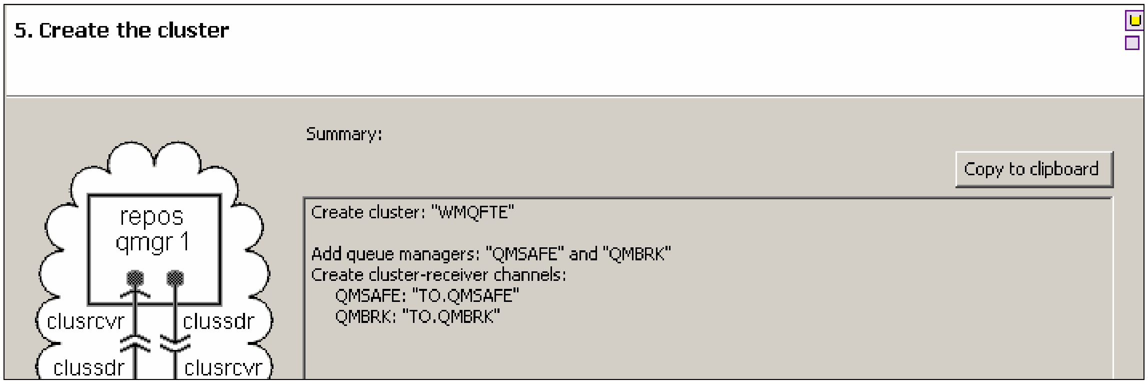
Cluster-receiver channel connection name:

sysd(14015)

Figure A-9 Details of second full repository receiver channel

Click **Next**.

On the Create the cluster panel, you see a summary of the channels that will be automatically be created between the two full repository queue managers (Figure A-10).



Create cluster: "WMQFTE"

Add queue managers: "QMSAFE" and "QMBRK"
Create cluster-receiver channels:
QMSAFE: "TO.QMSAFE"
QMBRK: "TO.QMBRK"

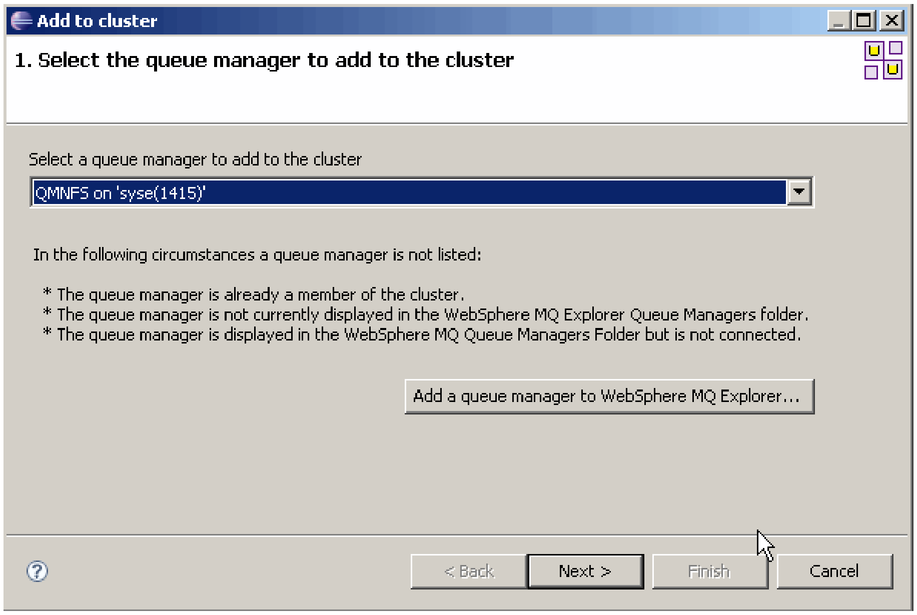
Figure A-10 Summary of channels between full repository queue managers

6. Click **Finish**.

We created the WMQFTE cluster with two full repository queue managers. Now we can add one or more queue managers to the cluster as partial queue managers.

For our system architecture, we have to add the queue manager QMNFS as a partial queue manager:

1. In the WebSphere MQ Explorer Navigator, on the left pane, under the Queue Manager Cluster tab, right-click **WMQFTE** and select **Add Queue Manager to the Cluster**.
2. On the panel for step 1, select the queue manager **QMNFS** from the drop-down menu, and then click **Next**.


Add to cluster
1. Select the queue manager to add to the cluster
Add a queue manager to WebSphere MQ Explorer...
< Back
Next >
Finish
Cancel

Select a queue manager to add to the cluster

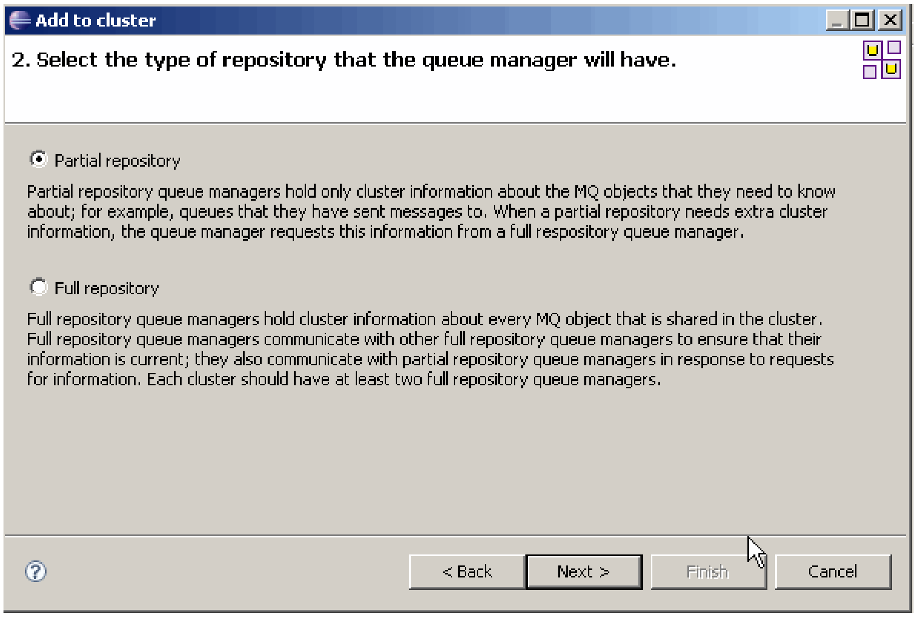
QMNFS on 'syse(1415)'

In the following circumstances a queue manager is not listed:

* The queue manager is already a member of the cluster.
* The queue manager is not currently displayed in the WebSphere MQ Explorer Queue Managers folder.
* The queue manager is displayed in the WebSphere MQ Queue Managers Folder but is not connected.

Figure A-11 Add new queue manager to the cluster

3. On the panel for step 2, check **Partial repository** and click **Next** (Figure A-12).



2. Select the type of repository that the queue manager will have.

● Partial repository

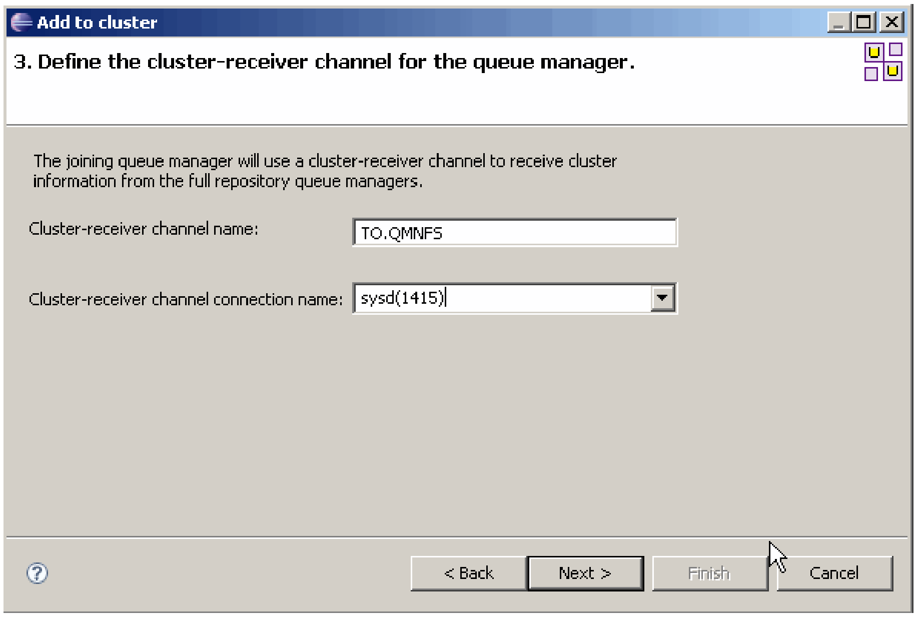
Partial repository queue managers hold only cluster information about the MQ objects that they need to know about; for example, queues that they have sent messages to. When a partial repository needs extra cluster information, the queue manager requests this information from a full respository queue manager.

○ Full repository

Full repository queue managers hold cluster information about every MQ object that is shared in the cluster. Full repository queue managers communicate with other full repository queue managers to ensure that their information is current; they also communicate with partial repository queue managers in response to requests for information. Each cluster should have at least two full repository queue managers.

Figure A-12 Define partial repository for the new queue manager

4. On the panel for step 3, enter the host name or IP address of the QMNFS queue manager machine, and then click **Next** (Figure A-13).



3. Define the cluster-receiver channel for the queue manager.

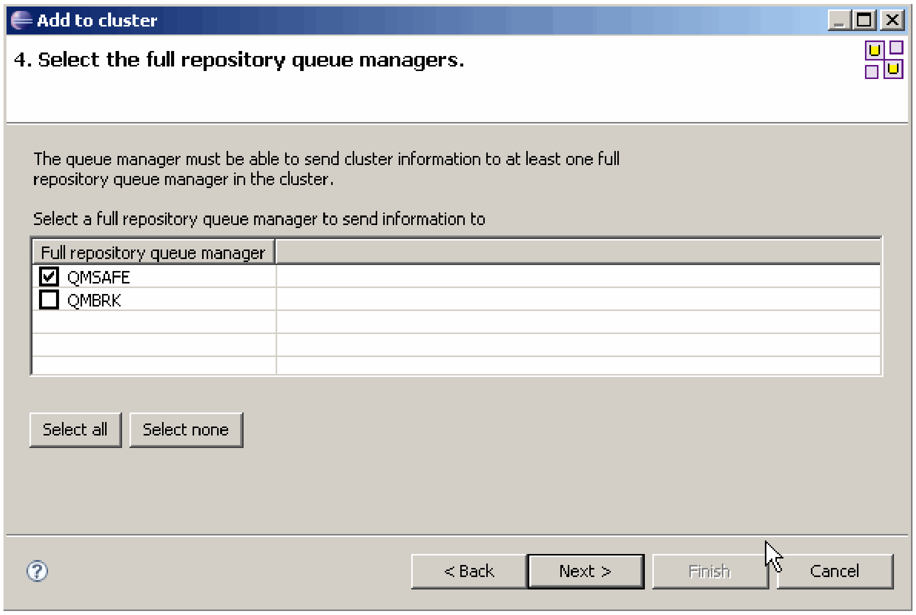
The joining queue manager will use a cluster-receiver channel to receive cluster information from the full repository queue managers.

Cluster-receiver channel name: TO.QMNFS

Cluster-receiver channel connection name: sysd(1415)

Figure A-13 Define cluster receiver channel for partial repository queue manager

5. On the panel for step 4, select a full repository queue manager and click **Next** (Figure A-14).



4. Select the full repository queue managers.

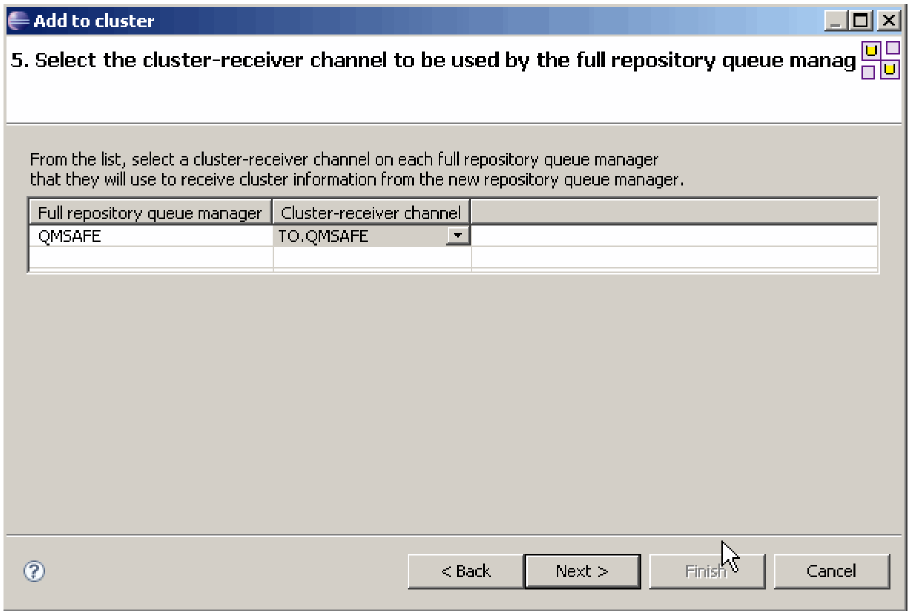
The queue manager must be able to send cluster information to at least one full repository queue manager in the cluster.

Select a full repository queue manager to send information to

Full repository queue manager	
☑ QMSAFE	
□ QMBRK	

Figure A-14 Select a full repository queue manager

6. On the panel for step 5, select the cluster-receiver channel of the full repository queue manager, and then click **Next** (Figure A-15).



5. Select the cluster-receiver channel to be used by the full repository queue manag

From the list, select a cluster-receiver channel on each full repository queue manager that they will use to receive cluster information from the new repository queue manager.

Full repository queue manager	Cluster-receiver channel	
QMSAFE	TO.QMSAFE	

Figure A-15 Define cluster receiver channel for full repository queue manager

7. On the next panel, click **Finish**.

You can ensure that the cluster is working correctly by checking the cluster sender and receiver channels of each member of the cluster. If the channels are up and running (you might have to start one or more of them), the cluster is set up correctly.

Creating the queue manager cluster using MQSC commands

Example A-4 is a sample mqsc script that defines queue manager QMSAFE as a repository queue manager for the queue manager cluster WMQFTE and creates cluster sender and receiver channels.

Example A-4 Define full repository queue manager and cluster channels

```
*--------------------------------------------------
* Setup Qmgr QMSAFE
*--------------------------------------------------
alter qmgr +
repos(WMQFTE)
*--------------------------------------------------
* Cluster Receiver Channel
*--------------------------------------------------
define channel(TO.QMSAFE) +
chltype(clusrcvr) +
cluster(WMQFTE) +
conname('sysd(14014)') +
replace

*--------------------------------------------------
* Cluster Sender Channel to QMBRK
*--------------------------------------------------
define channel(TO.QMBRK) +
chltype(clussdr) +
cluster(WMQFTE) +
conname('sysd(14015)') +
replace
```

You have to run the mqsc script for each queue manager that is a member of the cluster on its local machine to define the appropriate cluster sender and receiver channels.

Security

Because all of our scenarios are set in a secure network, we did not implement SSL security in the MQ FTE backbone. If there is a requirement to transmit sensitive information over their WebSphere MQ, enable Secure Socket Layer (SSL) security. Detailed information about how to set up SSL for WebSphere MQ V7 can be found at:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=/com.ibm.mq.csqzas.doc/sy10120_.htm

Configuring WebSphere MQ File Transfer Edition

In this section we show how to configure the WebSphere MQ File Transfer Edition components required by our scenarios.

Defining the coordination and command queue manager

This step can be done during the installation procedure of the WebSphere MQ File Transfer Edition Server package, with the precondition that you already created one or more queue managers, which then can be defined in the role of the coordination queue manager, command queue manager, or both.

We skipped this step in our WebSphere MQ File Transfer Edition Server installation procedure because we created our queue managers (QMSAFE, QMBRK, and QMNFS) in a subsequent step (as described in “Creating the queue managers” on page 271).

In the following steps, we define QMSAFE as the coordination queue manager and command queue manager for our MQ FTE backbone.

Creating the coordination queue manager

Open a command window and run the command shown in Example A-5 from the command line.

Example A-5 Define the coordination queue manager

```
fteSetupCoordination.cmd -coordinationQMgr QMSAFE
```

This command creates the following WebSphere MQ File Transfer Edition objects for queue manager QMSAFE:

- The config directory (`C:\IBM\WMQFTE\config\QMSAFE`)
- The `wmqfte.properties` file
- The `coordination.properties` file
- The `QMSAFE.mqsc` file

The `QMSAFE.mqsc` file provides the mqsc scripts that create the queues, topics, and namelists for the coordination queue manager. You must run these scripts against the queue manager QMSAFE from the command line using the command shown in Example A-6.

Example A-6 Create the WebSphere MQ objects

```
runmqsc QMSAFE < C:\IBM\WMQFTE\config\QMSAFE\QMSAFE.mqsc
```

Creating the command queue manager

For simplicity, queue manager QMSAFE combines roles for both the coordination queue manager and the command queue manager. In a command window we run the command shown in Example A-7.

Example A-7 Create the command queue manager

```
fteSetupCommands -connectionQMgr QMSAFE
```

This command creates the `command.properties` file in the coordination queue manager config directory (for example, `C:\IBM\WMQFTE\config\QMSAFE`).

Creating the WebSphere MQ File Transfer Edition agents

The MQ FTE backbone in the protected network for the file transfer scenarios includes four agents. The four agents send messages for publication to a single coordination queue manager. Figure A-16 depicts this architecture.

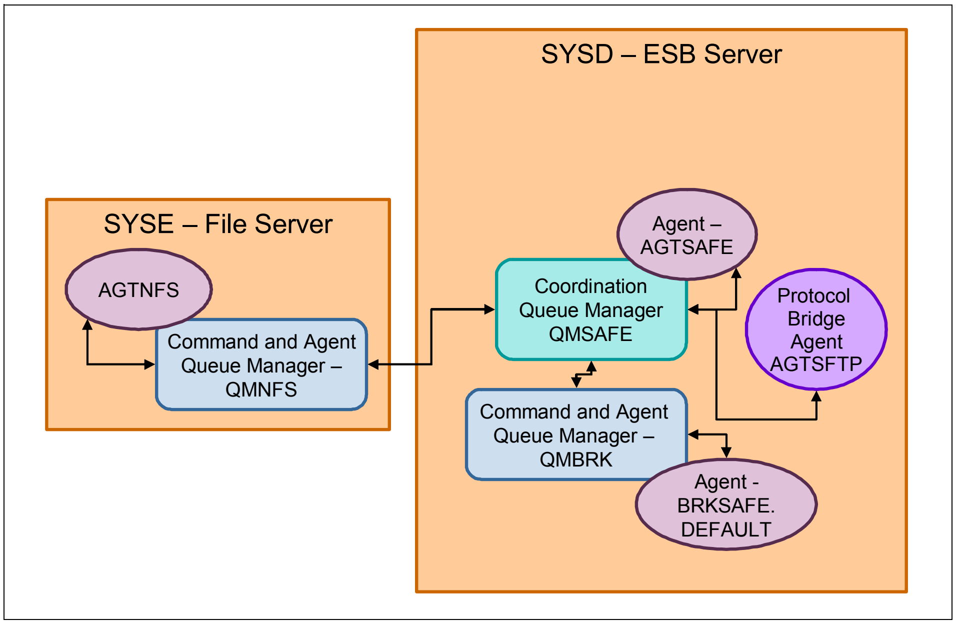

SYSD – ESB Server
SYSE – File Server
AGTNFS
Command and Agent Queue Manager – QMNFS
Agent – AGTSAFE
Coordination Queue Manager QMSAFE
Protocol Bridge Agent AGTSFTP
Command and Agent Queue Manager – QMBRK
Agent - BRKSAFE. DEFAULT

Figure A-16 Agents used in the scenarios

Agents used in the scenarios

This section provides a description of WebSphere MQ File Transfer Edition agents used in our scenario environment.

AGTNFS

This agent is used as a connection point between the B2B gateway and the MQ FTE backbone.

The B2B gateway writes incoming files to an Network File System (NFS) mount directory. AGTNFS then reads files from that directory and transfers them to the destination specified in the WebSphere MQ File Transfer Edition XML command received from the B2B gateway.

AGTNFS writes outgoing files to the NFS mount directory. The B2B gateway's NFS Poller Front Side Handler then reads the file from that directory and transfers the file to the destination specified by the partner destination associated with the partner specified in the file.

AGTNFS uses the queue manager, QMNFS, as its command and agent queue manager.

AGTSAFE

AGTSAFE is used to move files to and from back-end systems via an NFS directory. From this directory files can be either sent or received by the B2B gateway.

For outbound file transfers, a resource monitor is used by AGTSAFE to poll the local file system and start a file transfer to the NFS directory when a relevant file is found.

This agent uses the queue manager, QMSAFE, as its command and agent queue manager.

AGTSFTP

The AGTSFTP agent is a bridge agent. The protocol bridge allows our WebSphere MQ File Transfer Edition backbone (Figure 2-1 on page 17) to send and receive files stored on SFTP file servers. The creation and configuration of a bridge agent and how it is used in our overall architecture is described in detail Chapter 6, “Using FTP/SFTP with WebSphere MQ File Transfer Edition” on page 85.

This agent uses the queue manager, QMSAFE, as its command and agent queue manager.

BRKSAFE.DEFAULT

The BRKSAFE.DEFAULT agent is used for integrating WebSphere Message Broker brokers with WebSphere MQ File Transfer Edition agents in a WebSphere MQ File Transfer Edition backbone (Figure 2-1 on page 17). This agent runs in the WebSphere Message Broker execution group, default. It is created by the broker when a message flow with FTE nodes is deployed. The agent runs in a broker execution group. You do not need to create this agent manually. The configuration and usage of BRKSAFE.DEFAULT is documented in Chapter 8, “Integrating partner transfers with internal ESB” on page 175. BRKSAFE.DEFAULT uses the WebSphere MQ queue manager, QMBRK, as its command and agent queue manager.

How to create the AGTSAFE and AGTNFS agents

In this section, we describe the steps for creating and configuring the AGTSAFE and AGTNFS agents. These agents are used in a variety of scenarios in this book.

Creating agent AGTSAFE

We use the **fteCreateAgent** command (Example A-8) to create the AGTSAFE agent on sysd in binding mode to queue manager QMSAFE.

Example A-8 Creating AGTSAFE

```
ftecreateagent -agentname AGTSAFE -agentQMgr QMSAFE
```

The output from this command provides the following information about the objects that are created:

- The config directory of the AGTSAFE agent, located in the config directory of the agent queue manager QMSAFE:

  `C:\IBM\WMQFTE\config\QMSAFE\agents\AGTSAFE`

- An MQSC script that contains the MQSC commands to create all the system queues of the AGTSAFE agent, located in the config directory of the AGTSAFE agent.

> **For more information:** For more information about the **fteCreateAgent** command and a description of the valid parameters, see the WebSphere MQ File Transfer Edition Information Center at:
>
> http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/topic/com.ibm.wmqfte.admin.doc/create_agent_cmd.htm

Make sure that at the end of the output you see that the agent is successfully registered. If you see a message that the agent was configured but could not be registered, this means that the coordination queue manager could not be contacted because it is not available or your configuration parameters are not correct.

The effect is that the agent can be started and transfer files, but it is not listed by the **fteListAgents** command or in the WebSphere MQ File Transfer Edition Explorer. That means that you cannot define a transfer request in the WebSphere MQ File Transfer Edition Explorer using this agent, and furthermore that status messages of this agent are not shown in the WebSphere MQ File Transfer Edition Explorer Transfer Log view.

The WebSphere MQ reason code issued with the error provides more information about the reason for that problem. Explanations for reason codes can be found in the WebSphere MQ V7 Information Center at:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp

Creating the queues for the agent

As the next step, we create the agent's queues at the agent queue manager QMSAFE, using the MQSC script `AGTSAFE\AGTSAFE_create.mqsc`.

At a command prompt, enter the command shown in Example A-9.

Example A-9 Creating the agent queues

```
>runmqsc QMSAFE < C:\IBM\WMQFTE\config\QMSAFE\agents\AGTSAFE\AGTSAFE_create.mqsc
```

Starting the agent

Now we start the AGTSAFE agent using the **`fteStartAgent`** command (Example A-10).

Example A-10 fteStartAgent command

```
ftestartagent  AGTSAFE
```

You can check in the agent's log file whether the agent started successfully. Navigate to the log folder in the agent's config directory `C:\IBM\WMQFTE\config\QMSAFE\agents\AGTSAFE\log` and check the entry at the bottom of the `output0.log` file.

Creating the AGTNFS agent

We create the AGTNFS agent on syse using binding mode to connect to the local queue manager QMNFS. Because this agent uses QMSAFE on sysd as its coordination queue manager and command queue manager, we have to create the connection information for this queue manager prior to the creation of the agent.

Open a command window and enter the command shown in Example A-11.

Example A-11 Creating the connection information for the coordination queue manager

```
fteSetupCoordination -coordinationQMgr QMSAFE -coordinationQMgrHost sysd
-coordinationQMgrPort 14014
```

This command creates a `coordination.properties` file for queue manager QMSAFE in the WebSphere MQ File Transfer Edition config directory `C:\IBM\WMQFTE\config\QMSAFE`. This property file contains the connection information to the coordination queue manger QMSAFE, which is subsequently used with the **`fteCreateAgent`** command:

```
coordinationQMgr=QMSAFE
coordinationQMgrHost=sysd
coordinationQMgrChannel=SYSTEM.DEF.SVRCONN
coordinationQMgrPort=14014
```

Next, we enter the command in Example A-12 to create the command properties file with the connection information for the command queue manager.

Example A-12 Creating command properties file

```
fteSetupCommands -connectionQMgr QMSAFE -connectionQMgrHost sysd
-connectionQMgrPort 14014
```

This command creates a `command.properties` file for queue manager QMSAFE in the WebSphere MQ File Transfer Edition config directory `C:\IBM\WMQFTE\config\QMSAFE`. This property file contains the connection information for the command queue manger QMSAFE.

```
connectionQMgr=QMSAFE
connectionQMgrHost=sysd
connectionQMgrChannel=SYSTEM.DEF.SVRCONN
connectionQMgrPort=14014
```

We have defined all the required properties. The command in Example A-13 creates the AGTNFS agent in binding mode to the local queue manager QMNFS, with QMSAFE as the coordination manager.

Example A-13 Creating the AGTNFS agent

```
ftecreateagent -agentname AGTNFS -agentQMgr QMNFS -p QMSAFE
```

The -p parameter defines QMSAFE as the coordination queue manager. The connection details of this coordination queue manager are obtained from the `coordination.properties` file.

The output from this command provides information about the objects that are created:

- The config directory of the AGTNFS agent, located in the config directory of the agent queue manager QMSAFE, `C:\IBM\WMQFTE\config\QMSAFE\agents\AGTNFS`
- An MQSC script that contains the MQSC commands to create all the system queues of the AGTNFS agent, located in the config directory of the AGTNFS agent

Make sure that at the end of the output you see that the agent is successfully registered.

Creating the queues for the agent

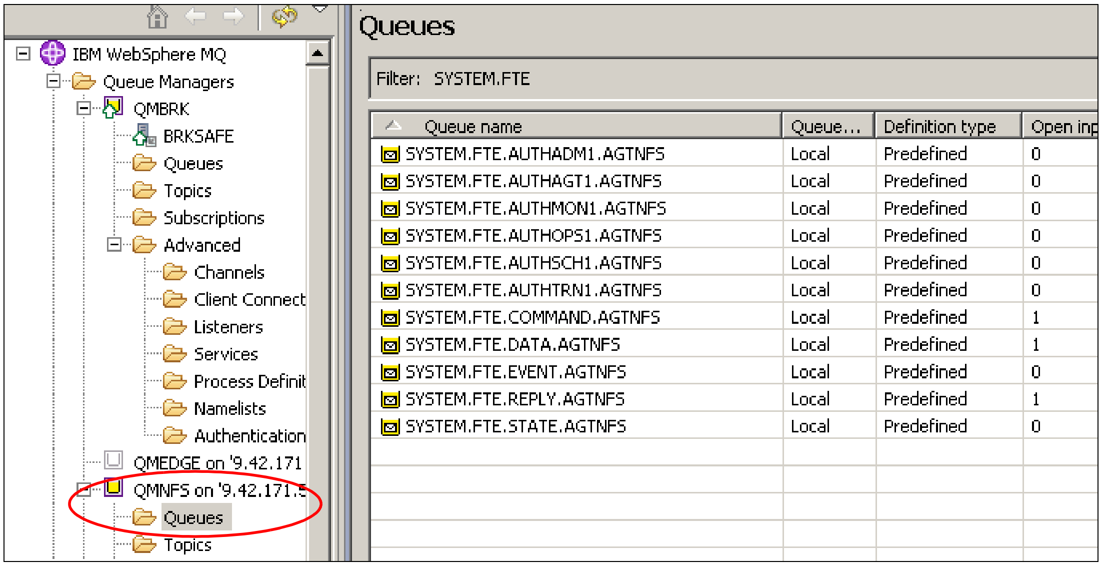
As the next step, we create the agent's queues at the agent queue manager QMNFS using the MQSC script `AGTNFS_create.mqsc`.

At a command prompt, enter the command shown in Example A-14.

Example A-14 Command

```
>runmqsc QMNFS < C:\IBM\WMQFTE\config\QMSAFE\agents\AGTNFS\AGTNFS_create.mqsc
```

You can see the system queues created by the command at the agent queue manager QMNFS. Open WebSphere MQ Explorer to make sure the that the SYSTEM.FTE.*fte_use.AGTNFS* queues are there (Figure A-17).



Queues

Filter: SYSTEM.FTE

Queue name	Queue...	Definition type	Open inp
SYSTEM.FTE.AUTHADM1.AGTNFS	Local	Predefined	0
SYSTEM.FTE.AUTHAGT1.AGTNFS	Local	Predefined	0
SYSTEM.FTE.AUTHMON1.AGTNFS	Local	Predefined	0
SYSTEM.FTE.AUTHOPS1.AGTNFS	Local	Predefined	0
SYSTEM.FTE.AUTHSCH1.AGTNFS	Local	Predefined	0
SYSTEM.FTE.AUTHTRN1.AGTNFS	Local	Predefined	0
SYSTEM.FTE.COMMAND.AGTNFS	Local	Predefined	1
SYSTEM.FTE.DATA.AGTNFS	Local	Predefined	1
SYSTEM.FTE.EVENT.AGTNFS	Local	Predefined	0
SYSTEM.FTE.REPLY.AGTNFS	Local	Predefined	1
SYSTEM.FTE.STATE.AGTNFS	Local	Predefined	0

Figure A-17 Agent queues

Starting the agent

Now we start the AGTNFS agent using the **`fteStartAgent`** command (Example A-15).

Example A-15 Starting the agent

```
ftestartagent  AGTNFS
```

You can check in the agent’s log file to see whether the agent started successfully. Navigate to the log folder in the agent’s config directory `C:\IBM\WMQFTE\config\QMSAFE\agents\AGTNFS\log` and check the entry at the bottom of the `output0.log` file.

Understanding WebSphere MQ File Transfer Edition properties files

WebSphere MQ File Transfer Edition uses several properties files that contain key information about your setup and are required for operation. The values defined in these properties files are used as the default parameters for all WebSphere MQ File Transfer Edition commands, unless you explicitly specify a different value on the command line.

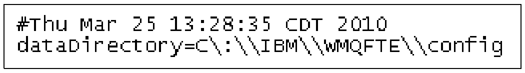
As an example, this section shows all of the properties that were set on sysd for the WebSphere MQ File Transfer Edition server and its AGTSAFE agent (see Figure A-16 on page 281). This includes:

- `install.properties`
- `wmqfte.properties`
- `command.properties`
- `coordination.properties`
- `agent.properties`

install.properties

This file contains one entry—the path to the WebSphere MQ File Transfer Edition data directory. This is a parameter that was set during the install.

This properties file is located in the WebSphere MQ File Transfer Edition installation directory. On sysd, it is located in the C:\IBM\WMQFTE\ directory. Figure A-18 shows its contents.

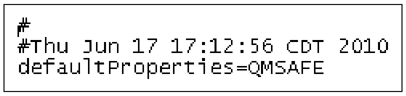


```
#Thu Mar 25 13:28:35 CDT 2010
dataDirectory=C\:\\IBM\\WMQFTE\\config
```

Figure A-18 install.properties

wmqfte.properties

This file contains one entry—the name of the default coordination queue manager. This properties file is located in the `C:\IBM\WMQFTE\config` directory. Figure A-19 shows its contents.



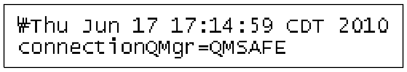
```
#
#Thu Jun 17 17:12:56 CDT 2010
defaultProperties=QMSAFE
```

Figure A-19 wmqfte.properties

command.properties

This file specifies the primary queue manager to connect to when issuing WebSphere MQ File Transfer Edition commands. (Certain commands also connect to the coodination queue manager.)

This properties file is located in the `C:\IBM\WMQFTE\config\QMSAFE` directory. Figure A-20 shows its contents.

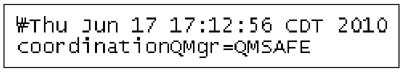


```
#Thu Jun 17 17:14:59 CDT 2010
connectionQMgr=QMSAFE
```

Figure A-20 command.properties

coordination.properties

This file specifies the coodination queue manager connection details. It is located in the `C:\IBM\WMQFTE\config\QMSAFE` directory. Figure A-21 shows its contents.

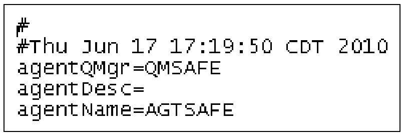


```
#Thu Jun 17 17:12:56 CDT 2010
coordinationQMgr=QMSAFE
```

Figure A-21 coordination.properties

agent.properties

This properties file is associated with a particular agent (for example, AGTSAFE). It is located in the `C:\IBM\WMQFTE\config\QMSAFE\agents\AGTSAFE` directory. Figure A-22 shows the contents of the `agent.properties` file for AGTSAFE.



```
#
#Thu Jun 17 17:19:50 CDT 2010
agentQMgr=QMSAFE
agentDesc=
agentName=AGTSAFE
```

Figure A-22 Agent properties

There are more than 30 parameters that can be specified for an agent. The agent minimally contains the name of the agent and information about how it connects to its queue manager. In the example above, these are the agentName and agentQMgr parameters. The majority of the possible agent parameters are associated with how the agent runs. These parameters can be important for tuning the agent's performance.

Configuring the database logger

The database logger is an optional component of WebSphere MQ File Transfer Edition that copies the published information from agents regarding transfers to a database. The database logger is a stand-alone Java application and is available in WebSphere MQ File Transfer Edition Version 7.0.1 or later. As part of the code for WebSphere MQ File Transfer Edition Remote Tools and Documentation, it can be configured for use on any platform where the WebSphere MQ File Transfer Edition Remote Tools and Documentation is licensed.

The database logger requires a DB2 or Oracle database as a prerequisite. In this scenario we use a DB2 V9.5, which must be installed prior to the setup that we describe in this section.

For information about how to configure the database logger, see the WebSphere MQ File Transfer Edition Information Center:

http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/topic/com.ibm.wmqfte.admin.doc/database_logger.htm

Installing the database logger

This section provides the procedure for installing the database required by the database logger. This example assumes a DB2 database on a Windows system.

Creating the database

To create the database:

1. Open a DB2 command line processor window. Select **Start** → **All Programs** → **BM DB2** → **DB2COPY1(default)** → **Command Line Tools** → **Command Line Processor**.
2. With this command line processor we can enter DB2-specific commands in a DB2 command mode. At the prompt, enter the following command to create the FTEAUDIT database:

   ```
   create database FTEAUDIT
   ```

3. Enter the following command to connect to the new database:

   ```
   connect to FTEAUDIT
   ```

4. Enter `quit` to exit the DB2 command mode.

   Example A-16 shows this series of commands and the results.

Example A-16 Commands to create the database

```
db2 => create database FTEAUDIT
DB20000I  The CREATE DATABASE command completed successfully.
db2 => connect to FTEAUDIT

   Database Connection Information

 Database server        = DB2/NT 9.5.1
```

```
 SQL authorization ID   = ADMIN
 Local database alias   = FTEAUDIT

db2 => quit
DB20000I  The QUIT command completed successfully.
```

5. At the command prompt, enter the following command to create the predefined tables for WebSphere MQ File Transfer Edition:

```
db2 -t -f "C:\ProgramFiles\IBM\WMQFTE\tools\sql\ftelog_tables_db2.sql"
```

6. Enter the following command to we grant access to the FTEAUDIT database for the user MUSR_MQADMIN:

```
db2 GRANT CONNECT ON DATABASE TO USER MUSR_MQADMIN
```

7. To store the transfer logs reliably in the database, we have to set up WebSphere MQ and DB2 to insert the MQ messages into the database under transaction control.

   Enter the following command to configure transaction support:

```
db2 UPDATE DBM CFG USING TP_MON_NAME MQ
```

   Close the DB2 command window.

Setting up WebSphere MQ for the logger

The next series of steps configures WebSphere QM for the database logger.

1. Enable WebSphere MQ as a transaction coordinator by copying the <*mq_install*>\java\lib\jdbc\jdbcdb2.dll file to the <*mq_install*>\exits directory.

   In our environment we copy the C:\IBM\MQ701\java\lib\jdbcjdbcdb2.dll file to the C:\IBM\MQ701\exits directory.

   This enables the queue manager to find and execute this file at startup.

2. Set the queue manager properties:

   a. Start the WebSphere MQ Explorer. In the navigation view on the right pane, right-click the coordination queue manager **QMSAFE** and select **Properties**.

   b. On the left pane of the properties list, select **XA resource managers**.

   c. Click **Add** (Figure A-23).

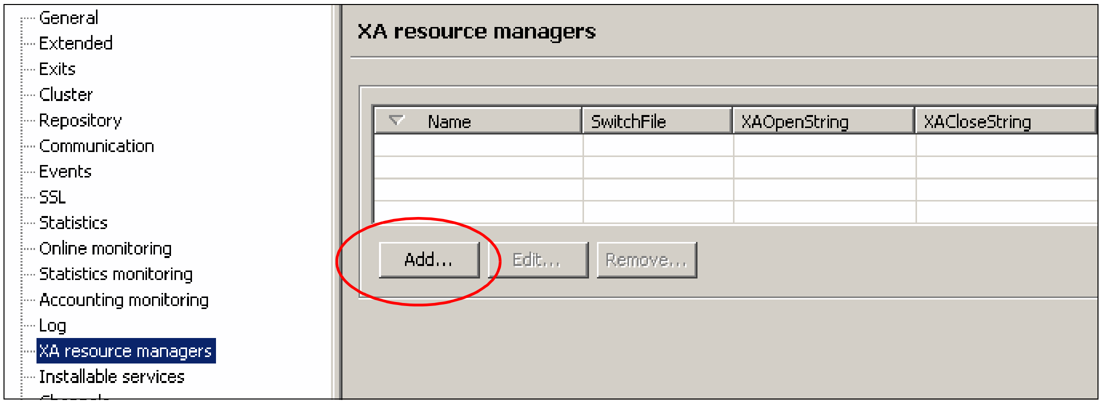
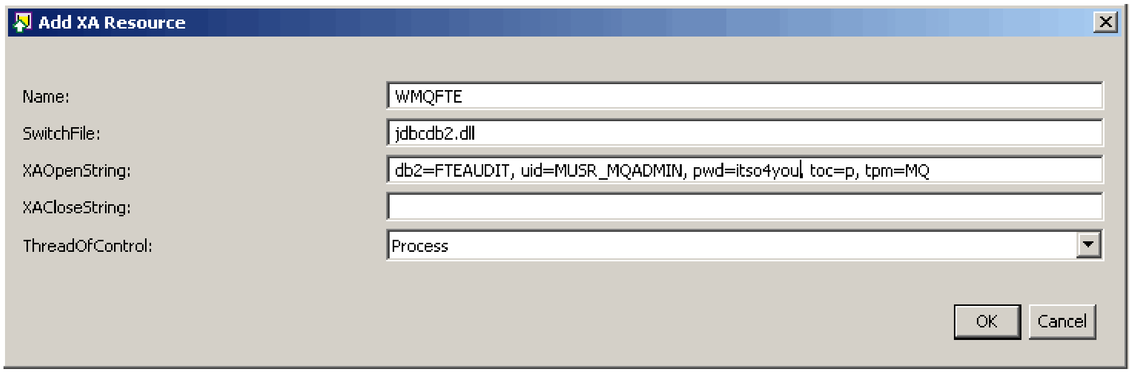


Figure A-23   Add an XA resource manager

d. Enter the following values:

- Name: `WMQFTE`
- SwitchFile: `jdbcdb2.dll`
- XAOpenString: `db2=FTEAUDIT, uid=MUSR_MQADMIN, pwd=`*`your_password`*`, toc=p, tpm=MQ`
- ThreadOfcontrol: `Process`

The XAOpenString contains the values required to access the database. FTEAUDIT is the database name, MUSR_MQADMIN is the user ID, and *your_password* is the new password that you choose (you will assign this password to MUSR_MQADMIN in step 3).



Name: WMQFTE

SwitchFile: jdbcdb2.dll

XAOpenString: db2=FTEAUDIT, uid=MUSR_MQADMIN, pwd=itso4you, toc=p, tpm=MQ

XACloseString:

ThreadOfControl: Process

Figure A-24   XA resource properties

e. Click **OK** to add these values, then click **Apply** and **OK** to close the QMSAFE properties.

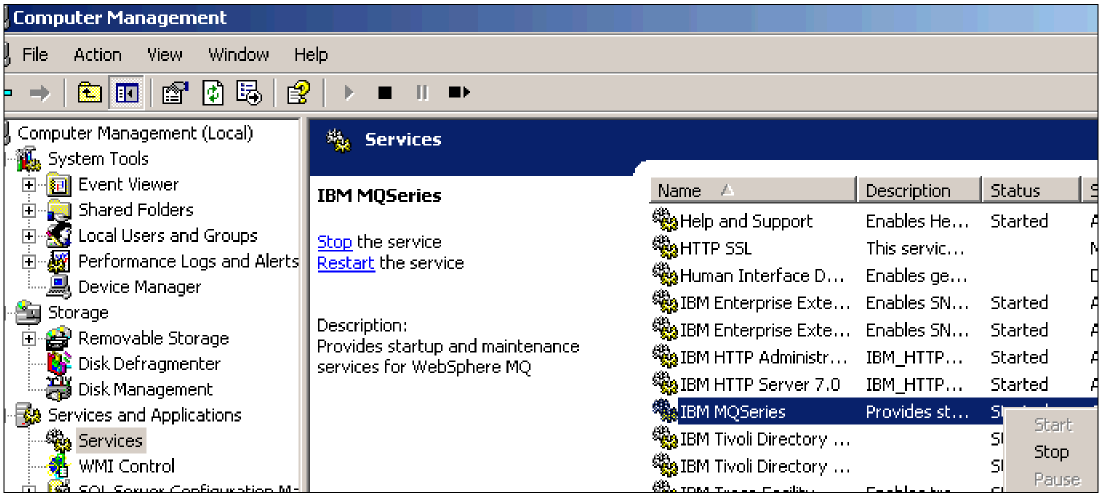
f. Close the WebSphere MQ Explorer.

3. Set the new password for user MUSR_MQADMIN.

a. Stop the IBM MQSeries® Service.

b. Click **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Computer Management**.

c. On the left side, under Services and Applications, select **Services**. Right-click **IBM MQSeries** and select **Stop** (Figure A-25).



IBM MQSeries

Stop the service
Restart the service

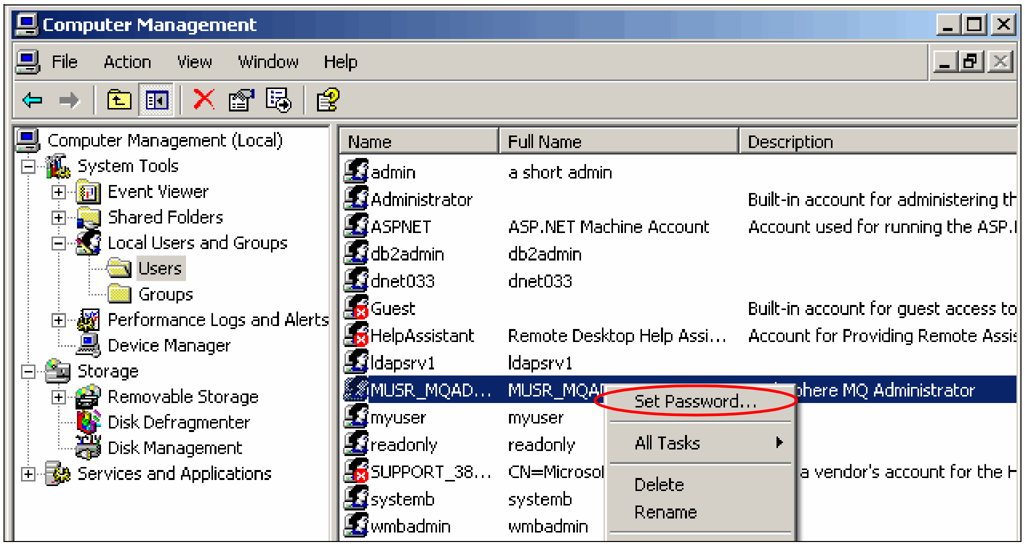
Description:
Provides startup and maintenance services for WebSphere MQ

Name	Description	Status
Help and Support	Enables He...	Started
HTTP SSL	This servic...	
Human Interface D...	Enables ge...	
IBM Enterprise Exte...	Enables SN...	Started
IBM Enterprise Exte...	Enables SN...	Started
IBM HTTP Administr...	IBM_HTTP...	Started
IBM HTTP Server 7.0	IBM_HTTP...	Started
IBM MQSeries	Provides st...	St
IBM Tivoli Directory ...		St
IBM Tivoli Directory ...		St

Figure A-25   Stop the IBM MQSeries service

d. Open **System Tools** → **Local Users and Groups** → **Users**.

e. Right-click **MUSR_MQADMIN** in the Name column and select **Set Password** (Figure A-26).

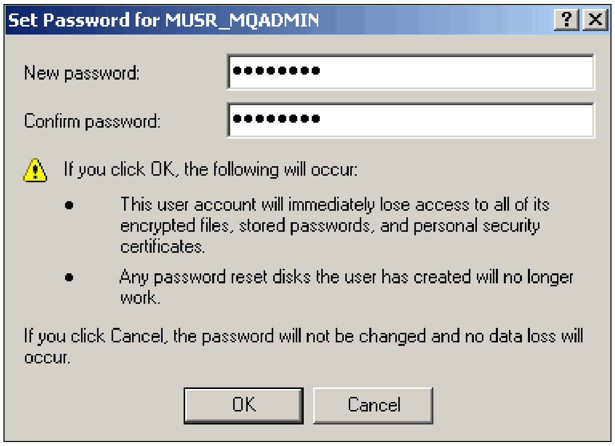


Name	Full Name	Description
admin	a short admin	
Administrator		Built-in account for administering th
ASPNET	ASP.NET Machine Account	Account used for running the ASP.I
db2admin	db2admin	
dnet033	dnet033	
Guest		Built-in account for guest access to
HelpAssistant	Remote Desktop Help Assi...	Account for Providing Remote Assis
ldapsrv1	ldapsrv1	
MUSR_MQAD...	MUSR_MQA	phere MQ Administrator
myuser	myuser	
readonly	readonly	
SUPPORT_38...	CN=Microsol	a vendor's account for the H
systemb	systemb	
wmbadmin	wmbadmin	

Figure A-26 Set the password for the user

f. Click **Proceed** on the next panel.

g. Enter the new password <*your_password*> for the user MUSR_MQADMIN and click **OK** (Figure A-27).



New password: ●●●●●●●●●

Confirm password: ●●●●●●●●●

If you click OK, the following will occur:

- This user account will immediately lose access to all of its encrypted files, stored passwords, and personal security certificates.
- Any password reset disks the user has created will no longer work.

If you click Cancel, the password will not be changed and no data loss will occur.

Figure A-27 Enter the new password

h. Right-click the user **MUSR_MQADMIN** again, select **Properties**, and select the **Member Of** tab (Figure A-28).

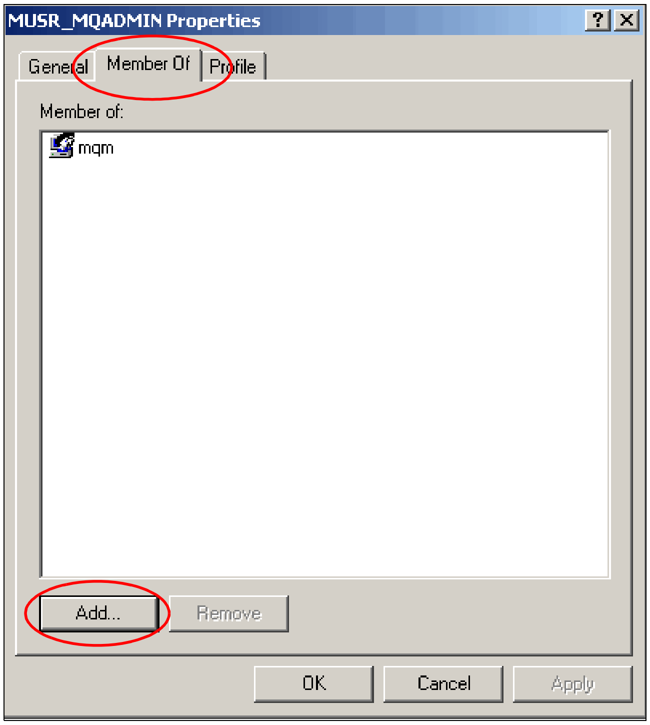
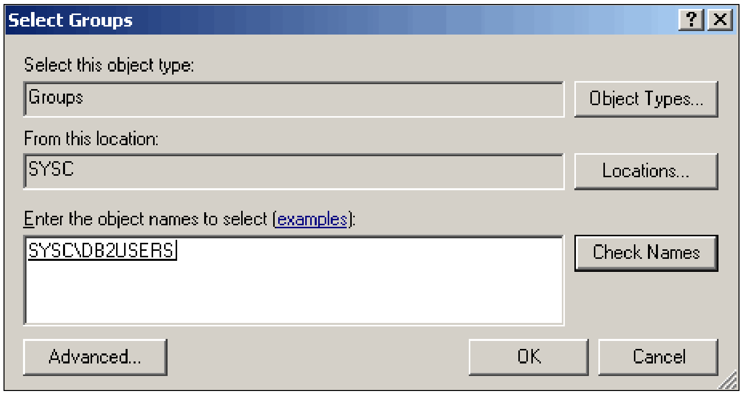


Figure A-28 Add the user to a group

Click **Add**.

i. Enter `DB2USERS` as the object name and click **Check Names** (Figure A-29). The host name should appear in front of the group name, which confirms that you have selected an existing group. (DB2USERS is a default group name created when you install DB2.)



Select this object type:
Groups

From this location:
SYSC

Enter the object names to select (examples):
SYSC\DB2USERS

Figure A-29 Select the DB2USERS group

Click **Ok**.

j. Close the Computer Management window.

4. Set the password for the IBM MQSeries Service COM server.

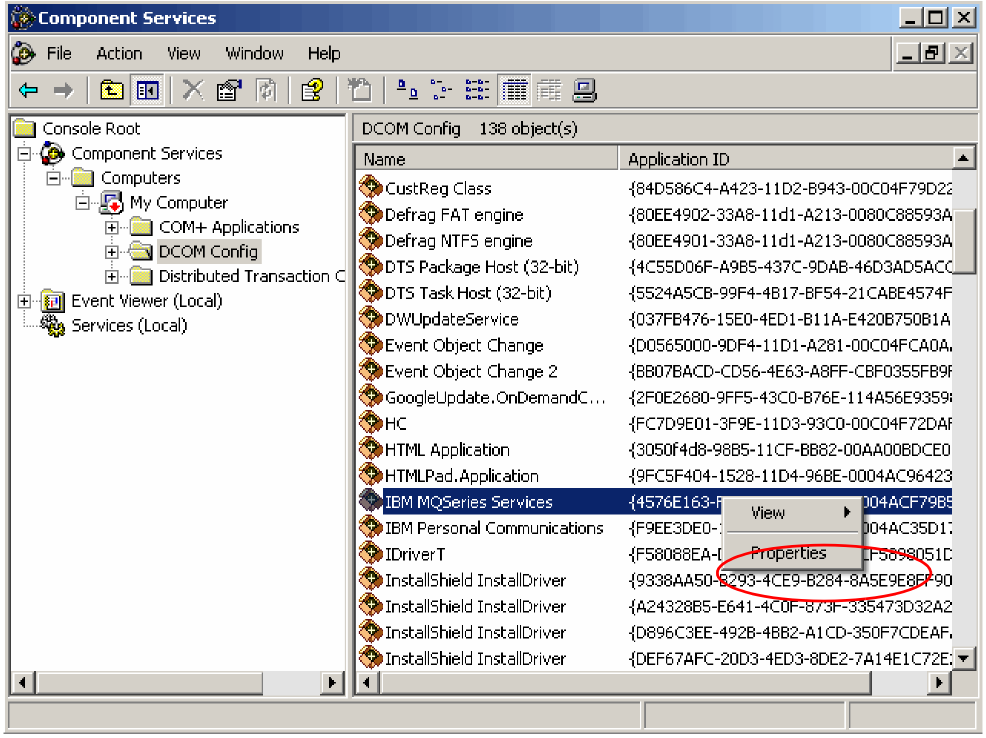
a. Open a command window and enter the command:

```
DCOMCNFG.EXE
```

b. On the left component select **Services** → **Computers** → **My computer** → **DCOM Config**.

c. Click **Yes** on the next three upcoming DCOM Configuration Warning windows.

d. Right-click **IBM MQSeries Services** and select **Properties** (Figure A-30).

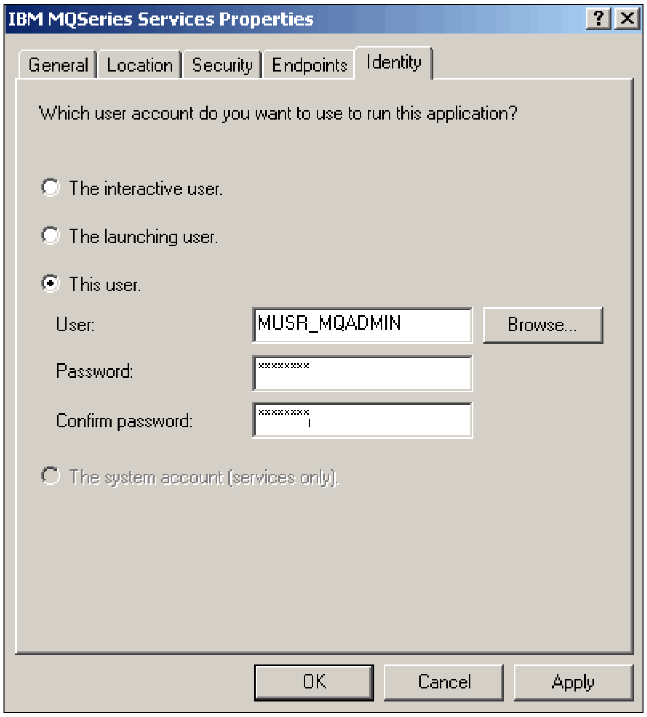


DCOM Config 138 object(s)

Name	Application ID
CustReg Class	{84D586C4-A423-11D2-B943-00C04F79D22
Defrag FAT engine	{80EE4902-33A8-11d1-A213-0080C88593A
Defrag NTFS engine	{80EE4901-33A8-11d1-A213-0080C88593A
DTS Package Host (32-bit)	{4C55D06F-A9B5-437C-9DAB-46D3AD5ACC
DTS Task Host (32-bit)	{5524A5CB-99F4-4B17-BF54-21CABE4574F
DWUpdateService	{037FB476-15E0-4ED1-B11A-E420B750B1A
Event Object Change	{D0565000-9DF4-11D1-A281-00C04FCA0A
Event Object Change 2	{BB07BACD-CD56-4E63-A8FF-CBF0355FB9F
GoogleUpdate.OnDemandC...	{2F0E2680-9FF5-43C0-B76E-114A56E9359
HC	{FC7D9E01-3F9E-11D3-93C0-00C04F72DAF
HTML Application	{3050f4d8-98B5-11CF-BB82-00AA00BDCE0
HTMLPad.Application	{9FC5F404-1528-11D4-96BE-0004AC96423
IBM MQSeries Services	{4576E163-F 004ACF79B5
IBM Personal Communications	{F9EE3DE0-1 004AC35D1
IDriverT	{F58088EA-D CF5898051D
InstallShield InstallDriver	{9338AA50-B293-4CE9-B284-8A5E9E8FF90
InstallShield InstallDriver	{A24328B5-E641-4C0F-873F-335473D32A2
InstallShield InstallDriver	{D896C3EE-492B-4BB2-A1CD-350F7CDEAF
InstallShield InstallDriver	{DEF67AFC-20D3-4ED3-8DE2-7A14E1C72E

Figure A-30 Open the IBM MQSeries Services properties

e. Select the **Identity** tab and enter the new password twice. This is the <*your_password*> value that was specified in the XAOpenString (Figure A-24 on page 289).



IBM MQSeries Services Properties

Which user account do you want to use to run this application?

○ The interactive user.

○ The launching user.

● This user.

User: MUSR_MQADMIN

Password: ********

Confirm password: ********

○ The system account (services only).

Figure A-31 Enter the new password

f. Click **Apply** → **OK** and close the Component Services window.

5. Modify the database logger properties.

You can use the sample `databaselogger.properties` file, which is provided in the <`config`> directory or create a new file with the name `databaselogger.properties` in the <`config`> directory of our queue manager `QMSAFE C:\IBM\WMQFTE\config\QMSAFE` and add the following lines:

```
wmqfte.queue.manager=QMSAFE
wmqfte.database.name=FTEAUDIT
wmqfte.database.driver=<SQLLIB_install_dir>/java/db2jcc.jar;
```

If you modify the sample `databaselogger.properties` file, the file should look like Example A-17.

Example A-17 Database logger properties

```
# ===============================================================
# @start_non_restricted_prolog@
# Version: %Z% %I% %W% %E% %U% [%H% %T%]
#
# Licensed Materials - Property of IBM
#
# 5655-U80, 5724-R10
#
# Copyright IBM Corp. 2009 All Rights Reserved.
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with
# IBM Corp.
# @end_non_restricted_prolog@
# ===============================================================
#
# Database Logger configuration properties
# ---------------------------------------
#
# Edit this file to configure the database logger according to
# your requirements. Commonly used properties are included in this
# sample file; see the InfoCenter for full documentation on
# all the properties available.
#
# Properties in this file are initially disabled by the inclusion of the #
# character at the beginning of the line. You should remove this
# character and then fill in the appropriate value after the = sign.

# The queue manager to which the database logger should connect.
# This must be a "bindings" connection to a queue manager on the
# same host, and will normally be your coordination queue manager.
# Example: wmqfte.queue.manager=COORD.QM

wmqfte.queue.manager=QMSAFE

# The name of the database which the database logger should use.
# You must create this database and import the FTE table structures
# before the database logger can run.
# Example: wmqfte.database.name=FTAUDIT1

wmqfte.database.name=FTEAUDIT

# The location of the JDBC driver jar file for your database.
# Multiple files may be provided, separated with ; characters on
# Windows and : characters on other platforms.
# Example: wmqfte.database.driver=/opt/IBM/db2/V9.5/java/db2jcc.jar

wmqfte.database.driver=C:\IBM\SQLLIB\java\db2jcc.jar

# The location of the native-code portion of your database driver,
# if any.
# Example: wmqfte.database.native.library.path=/opt/IBM/db2/V9.5/lib32/
#wmqfte.database.native.library.path=
```

6. Restart the IBM MQSeries service.
7. Start the database logger and view the results.

   We start the database logger using the FTESTARTDATABASELOGGER command in the command window (Example A-18).

Example A-18 Start the database logger

```
fteStartDatabaseLogger -F
5655-U80, 5724-R10 Copyright IBM Corp.  2008, 2009.  ALL RIGHTS RESERVED
BFGDB0023I: The database logger has completed startup activities and is now
running.
```

Note that we started the database logger with the -F option to run it in the foreground for testing purposes.

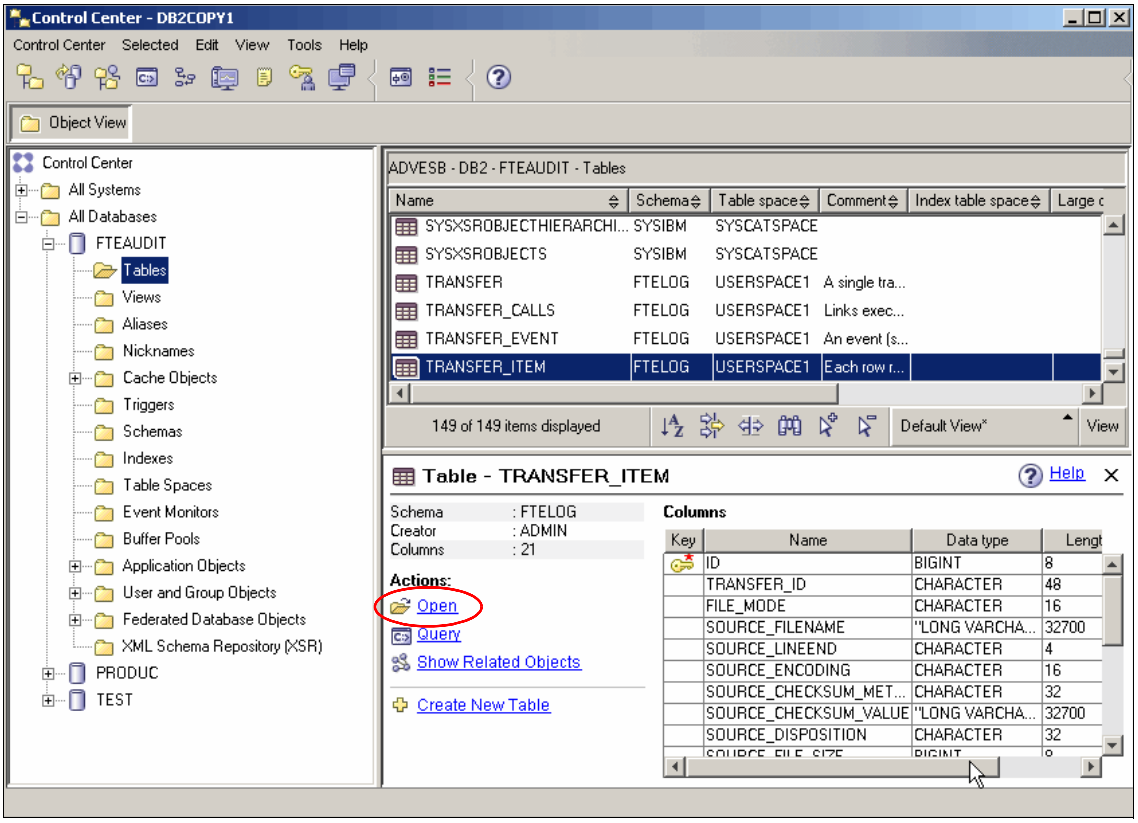
8. Execute one or more file transfers.

9. Now view the entries in the database.

   a. Start the DB2 control center. Click **Start** → **All Programs** → **IBM DB2** → **DB2COPY1** → **General Administration Tools** → **Control Center**.

   b. In the Control Center, select **All Databases** → **FTEAUDIT** → **Tables**.

   c. Select one of the logger tables (for example, TRANSFER_ITEM) and click **Open** (Figure A-32).



Name	Schema	Table space	Comment	Index table space	Large c
SYSXSROBJECTHIERARCHI...	SYSIBM	SYSCATSPACE			
SYSXSROBJECTS	SYSIBM	SYSCATSPACE			
TRANSFER	FTELOG	USERSPACE1	A single tra...		
TRANSFER_CALLS	FTELOG	USERSPACE1	Links exec...		
TRANSFER_EVENT	FTELOG	USERSPACE1	An event (s...		
TRANSFER_ITEM	FTELOG	USERSPACE1	Each row r...		

Table - TRANSFER_ITEM

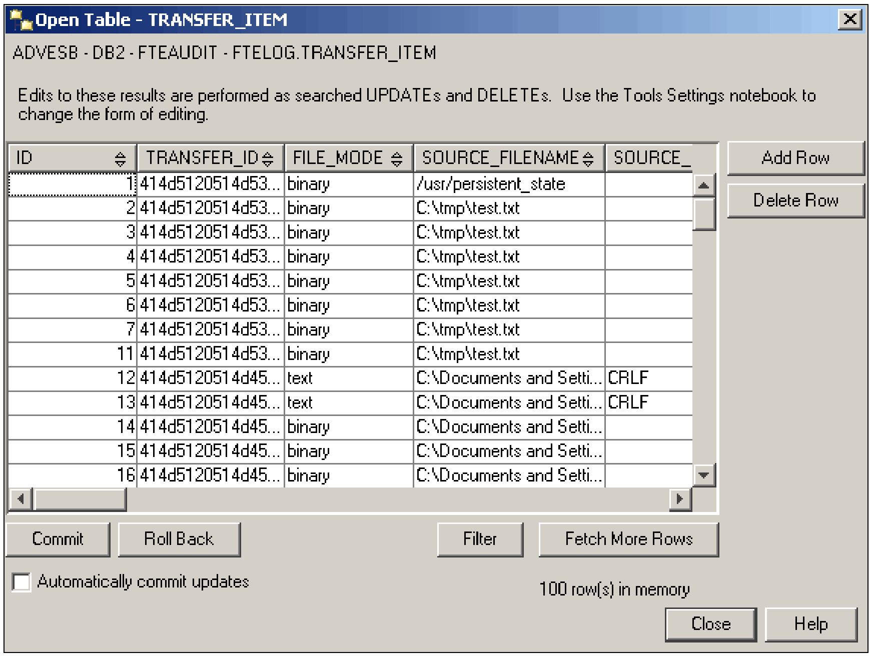
Schema : FTELOG
Creator : ADMIN
Columns : 21

Columns

Key	Name	Data type	Lengt
🔑	ID	BIGINT	8
	TRANSFER_ID	CHARACTER	48
	FILE_MODE	CHARACTER	16
	SOURCE_FILENAME	"LONG VARCHA...	32700
	SOURCE_LINEEND	CHARACTER	4
	SOURCE_ENCODING	CHARACTER	16
	SOURCE_CHECKSUM_MET...	CHARACTER	32
	SOURCE_CHECKSUM_VALUE	"LONG VARCHA...	32700
	SOURCE_DISPOSITION	CHARACTER	32
	SOURCE_FILE_SIZE	BIGINT	8

Figure A-32 Open the table

d. You should see now your first lines in the database (Figure A-33).



ADVESB - DB2 - FTEAUDIT - FTELOG.TRANSFER_ITEM

Edits to these results are performed as searched UPDATEs and DELETEs. Use the Tools Settings notebook to change the form of editing.

ID	TRANSFER_ID	FILE_MODE	SOURCE_FILENAME	SOURCE_
1	414d5120514d53...	binary	/usr/persistent_state	
2	414d5120514d53...	binary	C:\tmp\test.txt	
3	414d5120514d53...	binary	C:\tmp\test.txt	
4	414d5120514d53...	binary	C:\tmp\test.txt	
5	414d5120514d53...	binary	C:\tmp\test.txt	
6	414d5120514d53...	binary	C:\tmp\test.txt	
7	414d5120514d53...	binary	C:\tmp\test.txt	
11	414d5120514d53...	binary	C:\tmp\test.txt	
12	414d5120514d45...	text	C:\Documents and Setti...	CRLF
13	414d5120514d45...	text	C:\Documents and Setti...	CRLF
14	414d5120514d45...	binary	C:\Documents and Setti...	
15	414d5120514d45...	binary	C:\Documents and Setti...	
16	414d5120514d45...	binary	C:\Documents and Setti...	

100 row(s) in memory

Figure A-33 Logger entries

B

Preparing the WebSphere Application Server and IBM HTTP Server environment

Chapter 5, “Initiating file transfers outside your enterprise with HTTPS” on page 43, uses a WebSphere Application Server environment to host the WebSphere MQ File Transfer Edition Web Gateway and sample application supplied in the FO02 SupportPac. This scenario also uses an IBM HTTP Server as the first point of contact for requests entering the enterprise. This appendix provides information about how these environments were set up for use in this scenario.

WebSphere Application Server

The WebSphere MQ File Transfer Edition SupportPac FO02 runs on a J2EE application server. Any J2EE application server can be used. For our scenario, we installed WebSphere Application Server Network Deployment Version 7.0 and created a stand-alone application server to run the web gateway.

The application server was installed according to the directions in the WebSphere Application Server version 7.0 Information Center:

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/welc6topinstalling.html

After the application server is installed or if you are already running an application serving environment, review workload requirements to decide which application server topology best fits business needs. Review information in the information under “Setting up the application serving environment” in the WebSphere Application Server Version 7.0 Information Center:

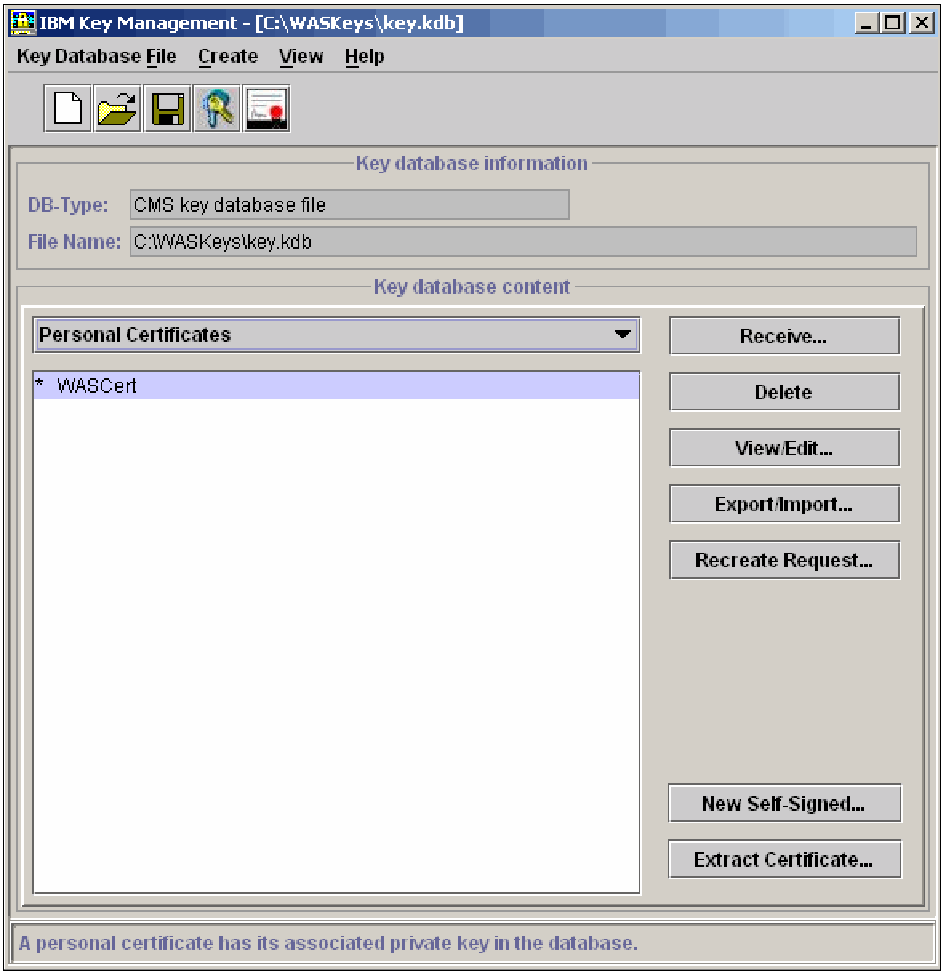
http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/welc6topserver.html

For this scenario, we configured a stand-alone application server in the Profile Management Tool for WebSphere Application Server (Figure B-2 on page 300). When building the application server environment, we created our own key store and certificates for the application server, disabled administrative security, and modified port numbers. The steps to accomplish these modifications are described in the following sections.

IBM Key Management

Before starting the Profile Management Tool to configure the WebSphere Application Server, create a key store and any personal certificates required for the application server. Refer to your organization’s security policies for the appropriate key management practices for your application serving environment.

To create the keystore and certificates for the application server in the scenario, the IBM Key Management tool is used. Figure B-1 shows the keystore and the personal certificate.



DB-Type: CMS key database file

File Name: C:\WASKeys\key.kdb

Personal Certificates

* WASCert

A personal certificate has its associated private key in the database.

Figure B-1 IBM Key Management

The IBM Key Management Tool is included with IBM HTTP Server. You can use the following commands to start it:

```
C:\HTTPServer\gsk7\bin>set JAVA_HOME=C:\HTTPServer\java\jre
C:\HTTPServer\gsk7\bin>gsk7ikm
```

More information about how to use the IBM Key Management tool can be found at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.ihs.doc/info/ihs/ihs/cihs_ikeycmdline.html

After security certificates and other practices are in place, it is time to use the Profile Management Tool to create the server environment.

Profile Management Tool 7.0

For this scenario, we used the stand-alone application server profile in the Profile Management Tool for WebSphere Application Server (Figure B-2).

> **Note for z/OS users:** If you are creating your application server environment on z/OS, see Techdoc document PRS3341 for more information about how to configure your environment:
>
> http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS3341

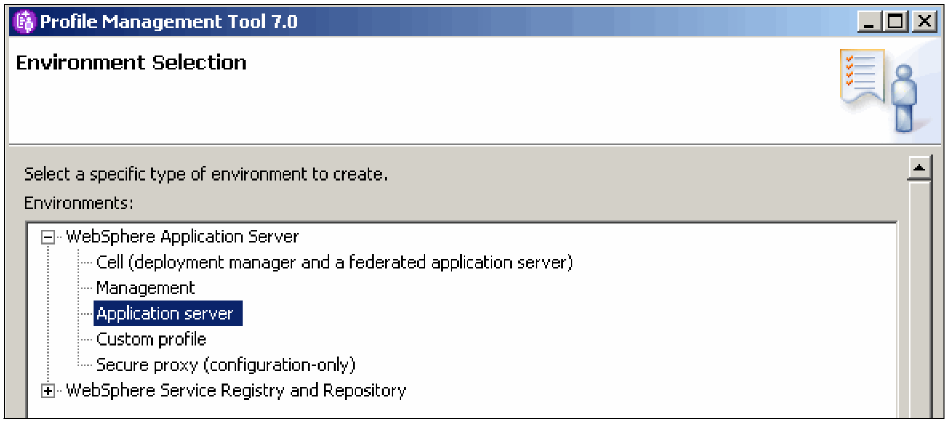
Follow the Profile Management Tool wizard panels to create your WebSphere Application Server environment. When working with the advanced profile creation in the Profile Management Tool, pay special attention to the security certificates that the application server will use, the ports that the application server is configured to use, and the types of security enabled. Refer to your organization's security practices for the appropriate security choices. For more information about the selections that can be made when configuring your WebSphere Application Server Environment, see the WebSphere Application Server version 7.0 Information Center:

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.xmlfep.multiplatform.doc/info/ae/ae/tpro_instancessaappserv.html

Creating the WebSphere Application Server environment

To create a stand-alone application server similar to what is used in our scenario, follow the steps below:

1. From the Profile Management Tool, create a stand-alone server by selecting **Application Server** (Figure B-2). Click **Next**.



Profile Management Tool 7.0

Environment Selection

Select a specific type of environment to create.

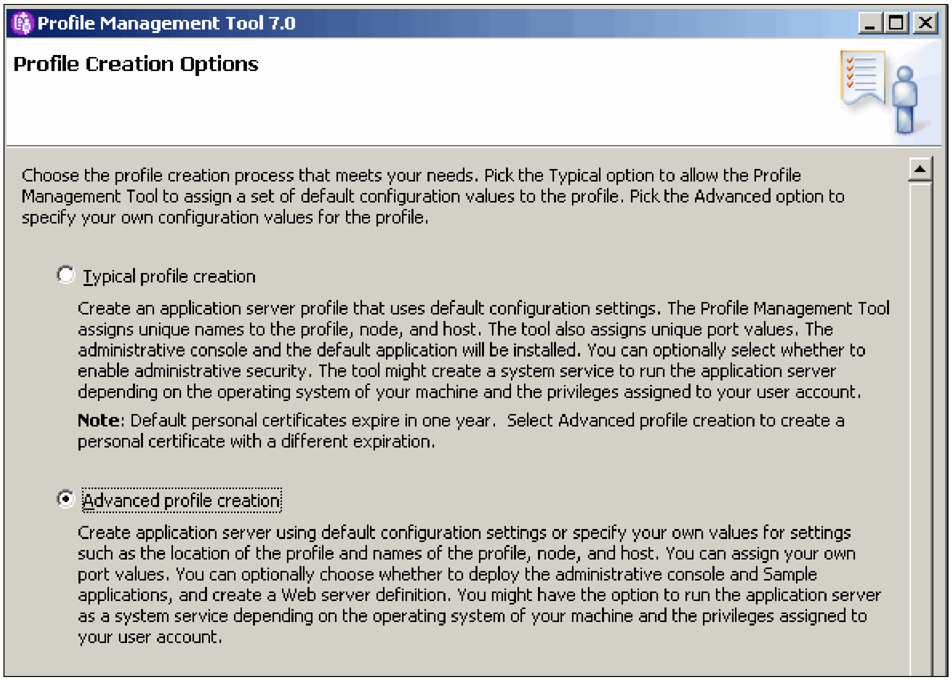
Environments:

- WebSphere Application Server
  - Cell (deployment manager and a federated application server)
  - Management
  - Application server
  - Custom profile
  - Secure proxy (configuration-only)
- WebSphere Service Registry and Repository

Figure B-2 Stand-alone application server profile in Profile Management Tool

2. Select **Advanced profile creation** for the profile creation option.

   In this scenario, we created our application server environment through the Advanced profile creation option (Figure B-3). This option allowed us to create our own application server name, configuration directory, and node name, and gave us the ability to make many more modifications. Most notably, we imported our own personal certificates for the application server and modified port numbers.



Profile Creation Options

Choose the profile creation process that meets your needs. Pick the Typical option to allow the Profile Management Tool to assign a set of default configuration values to the profile. Pick the Advanced option to specify your own configuration values for the profile.

○ Typical profile creation

Create an application server profile that uses default configuration settings. The Profile Management Tool assigns unique names to the profile, node, and host. The tool also assigns unique port values. The administrative console and the default application will be installed. You can optionally select whether to enable administrative security. The tool might create a system service to run the application server depending on the operating system of your machine and the privileges assigned to your user account.

Note: Default personal certificates expire in one year. Select Advanced profile creation to create a personal certificate with a different expiration.

● Advanced profile creation

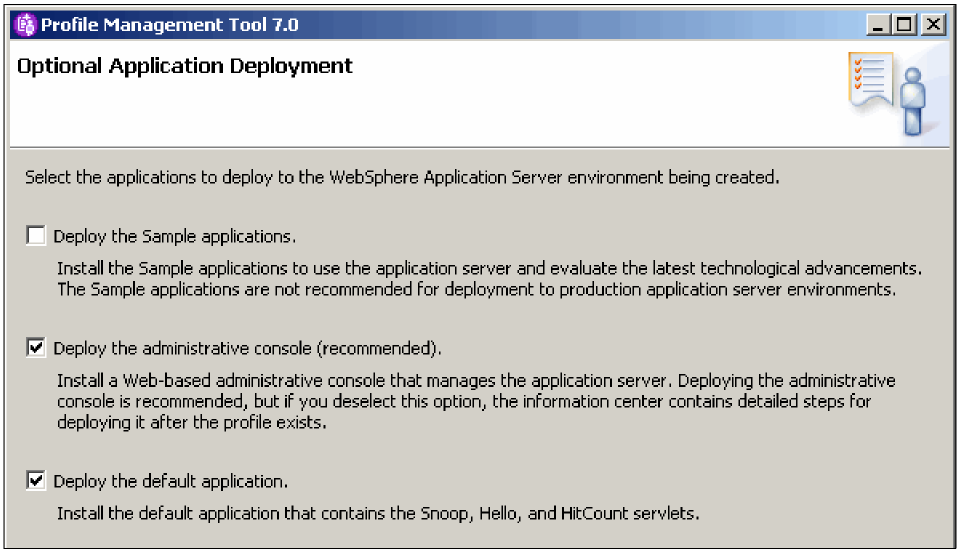
Create application server using default configuration settings or specify your own values for settings such as the location of the profile and names of the profile, node, and host. You can assign your own port values. You can optionally choose whether to deploy the administrative console and Sample applications, and create a Web server definition. You might have the option to run the application server as a system service depending on the operating system of your machine and the privileges assigned to your user account.

Figure B-3 Profile Management Tool - Advanced Profile Creation

   Click **Next**.

3. On the Optional Application Deployment page, choose the applications to be deployed with the application server environment.

   For our purposes, we elected to accept the default checked options (Figure B-4).



Optional Application Deployment

Select the applications to deploy to the WebSphere Application Server environment being created.

☐ Deploy the Sample applications.

Install the Sample applications to use the application server and evaluate the latest technological advancements. The Sample applications are not recommended for deployment to production application server environments.

☑ Deploy the administrative console (recommended).

Install a Web-based administrative console that manages the application server. Deploying the administrative console is recommended, but if you deselect this option, the information center contains detailed steps for deploying it after the profile exists.

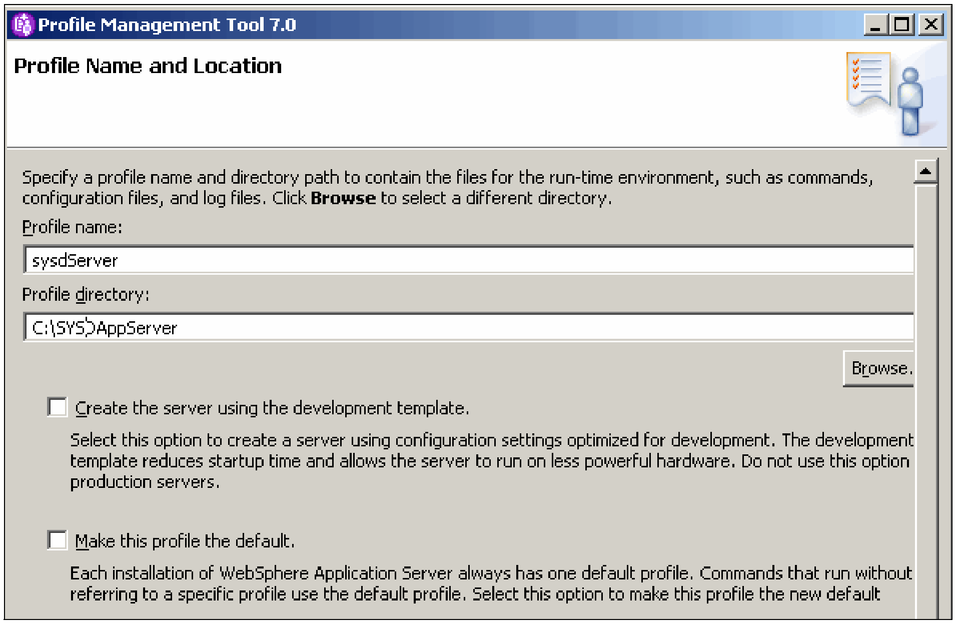
☑ Deploy the default application.

Install the default application that contains the Snoop, Hello, and HitCount servlets.

Figure B-4 Profile Management Tool - Optional Application Deployment

Click **Next**.

4. Create the server name, sysdServer, and profile directory (`C:\SYSDAppServer`) on the Profile Name and Location page (Figure B-5).



Profile Name and Location

Specify a profile name and directory path to contain the files for the run-time environment, such as commands, configuration files, and log files. Click **Browse** to select a different directory.

Profile name:

sysdServer

Profile directory:

C:\SYSDAppServer

□ Create the server using the development template.

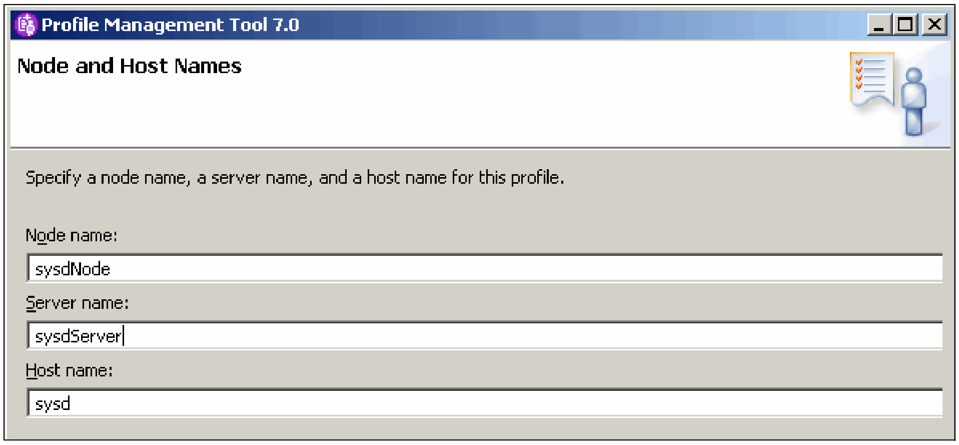
Select this option to create a server using configuration settings optimized for development. The development template reduces startup time and allows the server to run on less powerful hardware. Do not use this option production servers.

□ Make this profile the default.

Each installation of WebSphere Application Server always has one default profile. Commands that run without referring to a specific profile use the default profile. Select this option to make this profile the new default

Figure B-5 Profile Management Tool - Profile Name and Location

5. Enter the node and host names for your environment. In our environment we created the following names (Figure B-6).
   - Node name: `sysdNode`
   - Server name: `sysdServer`
   - Host name: `sysd`



Node and Host Names

Specify a node name, a server name, and a host name for this profile.

Node name:

sysdNode

Server name:

sysdServer

Host name:

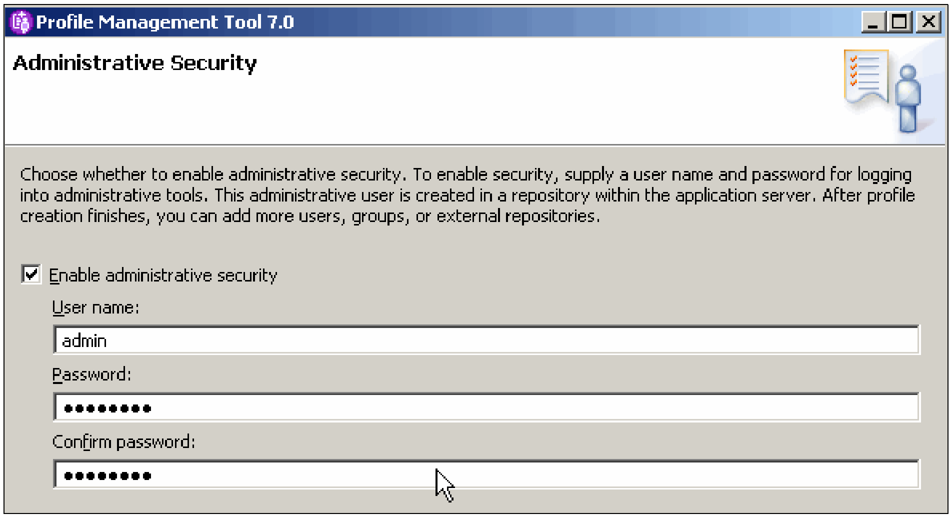
sysd

Figure B-6 Profile Management Tool - Node and Host Names

Click **Next**.

6. On the Administrative Security panel, enter the user ID and password that you want to use to log onto the Administrative Console after the application server is running.

After logging onto the Administrative Console, more user IDs and passwords can be added and configured. For our environment, we used the admin user ID and associated password. Figure B-7 shows this entry.



Administrative Security

Choose whether to enable administrative security. To enable security, supply a user name and password for logging into administrative tools. This administrative user is created in a repository within the application server. After profile creation finishes, you can add more users, groups, or external repositories.

☑ Enable administrative security

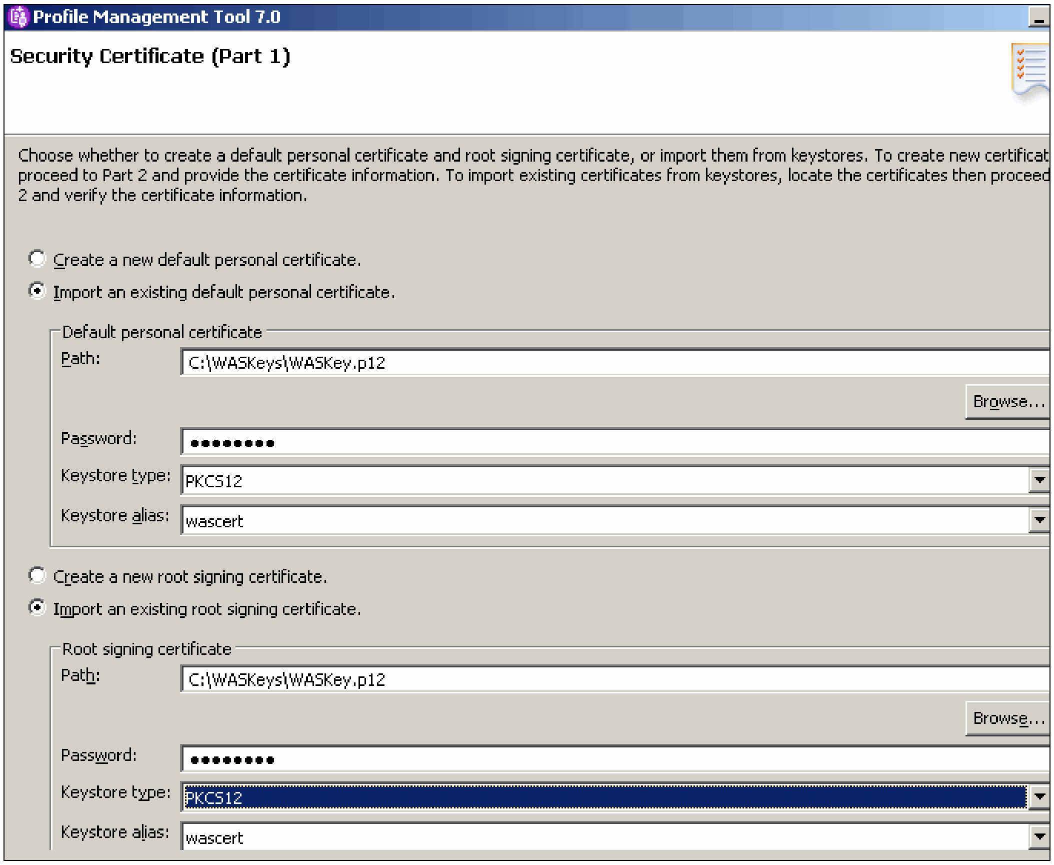
User name: admin

Password: ●●●●●●●●●

Confirm password: ●●●●●●●●●

Figure B-7 Profile Management Tool - Administrative Security

7. For Security Certificate (Part 1), import your organization's personal certificate or allow WebSphere Application Server to create one for you. In our scenario, we use the personal certificate created with IBM Key Management for the application server. Figure B-8 shows the selection of the certificate.



Security Certificate (Part 1)

Choose whether to create a default personal certificate and root signing certificate, or import them from keystores. To create new certificat proceed to Part 2 and provide the certificate information. To import existing certificates from keystores, locate the certificates then proceed 2 and verify the certificate information.

○ Create a new default personal certificate.

● Import an existing default personal certificate.

Default personal certificate

Path: C:\WASKeys\WASKey.p12

Password: ●●●●●●●●●

Keystore type: PKCS12

Keystore alias: wascert

○ Create a new root signing certificate.

● Import an existing root signing certificate.

Root signing certificate

Path: C:\WASKeys\WASKey.p12

Password: ●●●●●●●●●

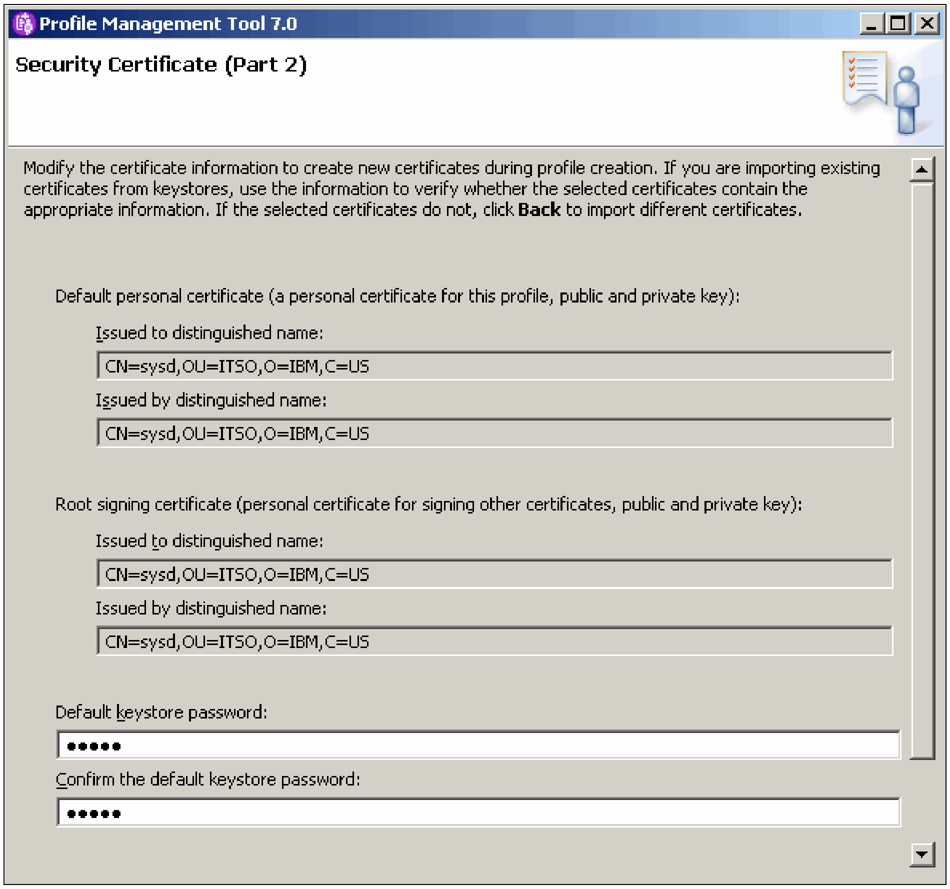
Keystore type: PKCS12

Keystore alias: wascert

Figure B-8 Profile Management Tools - Security Certificate (Part 1)

Click **Next**.

8. Use the Security Certificate (Part 2) panel to create the password for the default keystore for WebSphere Application Server (Figure B-9).



Profile Management Tool 7.0

Security Certificate (Part 2)

Modify the certificate information to create new certificates during profile creation. If you are importing existing certificates from keystores, use the information to verify whether the selected certificates contain the appropriate information. If the selected certificates do not, click **Back** to import different certificates.

Default personal certificate (a personal certificate for this profile, public and private key):

Issued to distinguished name:

CN=sysd,OU=ITSO,O=IBM,C=US

Issued by distinguished name:

CN=sysd,OU=ITSO,O=IBM,C=US

Root signing certificate (personal certificate for signing other certificates, public and private key):

Issued to distinguished name:

CN=sysd,OU=ITSO,O=IBM,C=US

Issued by distinguished name:

CN=sysd,OU=ITSO,O=IBM,C=US

Default keystore password:

●●●●●

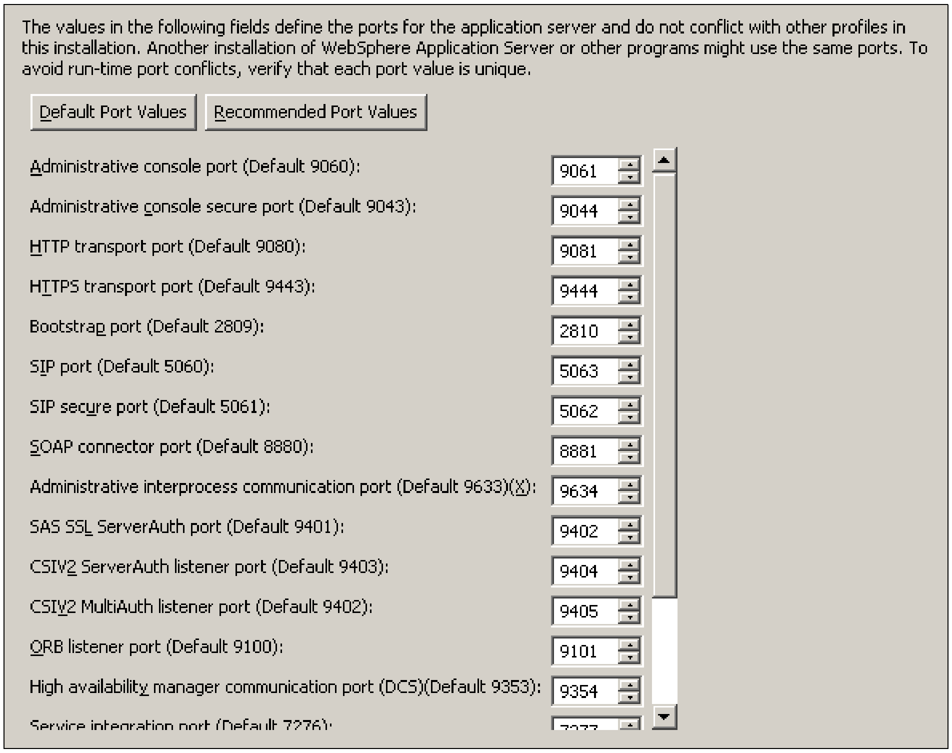
Confirm the default keystore password:

●●●●●

Figure B-9 Profile Management Tool - Security Certificate (Part 2)

Click **Next**.

9. Customize your port selections on the Port Values Assignment panel. Several of these ports might already be in use. Refer to your organization's policy for assigning ports. For our purposes, the ports are as listed in Figure B-10. Take note of the HTTP Port (9081) and HTTPS Port (9444) for use with the Web Gateway application.



The values in the following fields define the ports for the application server and do not conflict with other profiles in this installation. Another installation of WebSphere Application Server or other programs might use the same ports. To avoid run-time port conflicts, verify that each port value is unique.

Administrative console port (Default 9060):	9061
Administrative console secure port (Default 9043):	9044
HTTP transport port (Default 9080):	9081
HTTPS transport port (Default 9443):	9444
Bootstrap port (Default 2809):	2810
SIP port (Default 5060):	5063
SIP secure port (Default 5061):	5062
SOAP connector port (Default 8880):	8881
Administrative interprocess communication port (Default 9633)(X):	9634
SAS SSL ServerAuth port (Default 9401):	9402
CSIV2 ServerAuth listener port (Default 9403):	9404
CSIV2 MultiAuth listener port (Default 9402):	9405
ORB listener port (Default 9100):	9101
High availability manager communication port (DCS)(Default 9353):	9354
Service integration port (Default 7276):	7277

Figure B-10 Profile Management Tool - Port Values Assignment

Click **Next**.

10. On Windows, you will encounter the Windows Service Definition panel. Choose how to run your application server.

    For our scenario, we opted not to run the application server as a Windows service to help simplify the environment and allow more manual control over the application server environment.

    Click **Next**.

11. On the Web Server Definition panel, accept the default of not creating a web server definition. Click **Next**.

12. When you reach the Profile Creation Summary, review your configuration and select **Create**.

Starting and verifying the environment

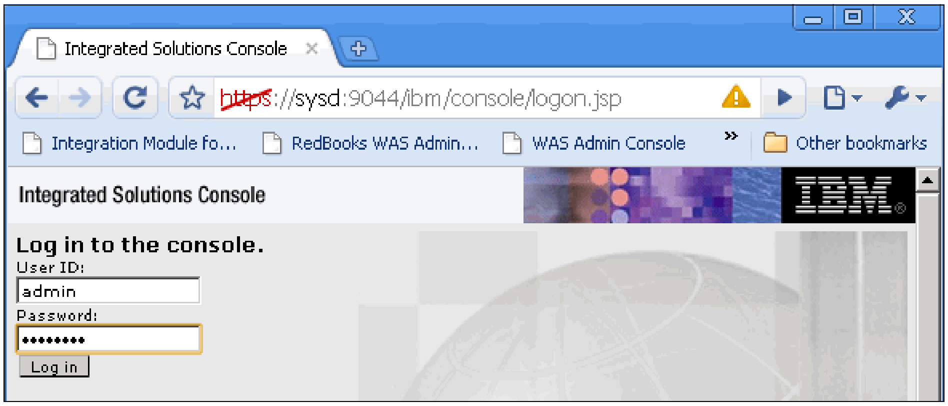
After the application server environment has been successfully created, start the application server. This can be done through menu options on Windows operating systems or through a command prompt. More dialog regarding the application server startup can be seen by issuing the start command from a terminal window. Example B-1 shows an example of how our application server is started.

Example: B-1 Start WebSphere Application Server stand-alone server

```
C:\SYSCDAppServer\bin>startServer.bat sysdServer
ADMU0116I: Tool information is being logged in file
           C:\SYSCDAppServer\logs\sysdServer\startServer.log
ADMU0128I: Starting tool with the SYSDAppServer profile
ADMU3100I: Reading configuration for server: sysdServer
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server sysdServer open for e-business; process id is 4224
```

When the application server is running, log onto the WebSphere Application Server console to make sure that the server is running and working (Figure B-11). The Administration Console can be accessed through the following URL:

```
http://<host>:<AdminConsolePort>/ibm/console
```


Integrated Solutions Console
~~https~~://sysd:9044/ibm/console/logon.jsp
Integration Module fo...
RedBooks WAS Admin...
WAS Admin Console
Other bookmarks
Integrated Solutions Console
IBM

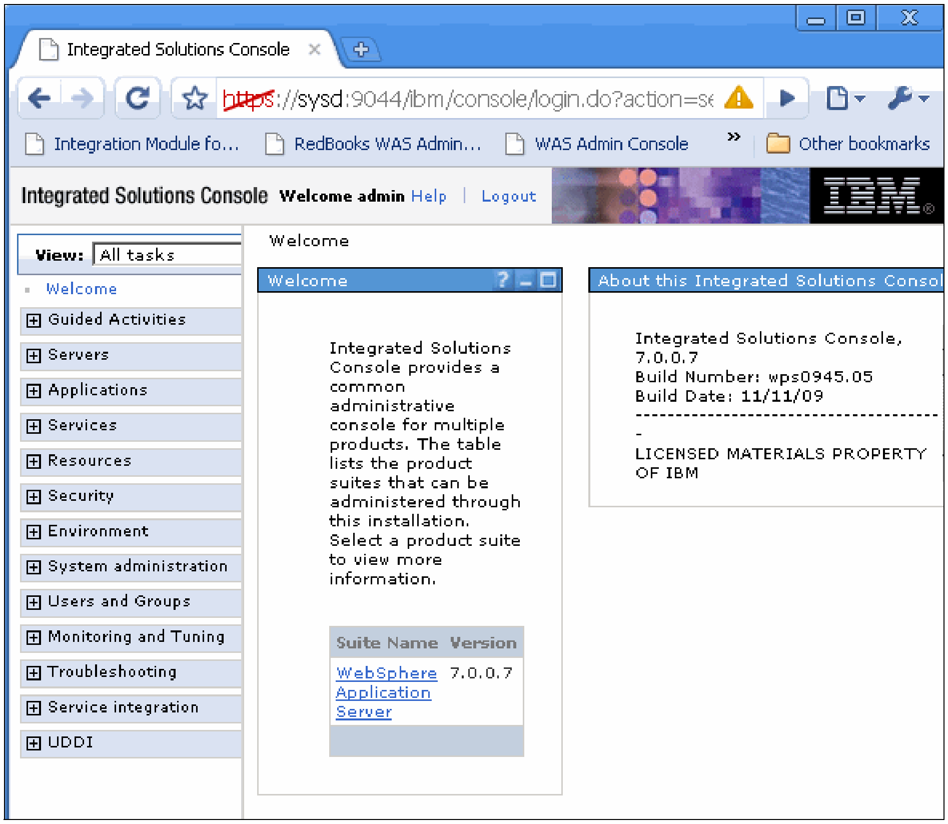
Log in to the console.

User ID: admin

Password: ••••••••

Figure B-11 WebSphere Application Server Administration Console login

13. After you have successfully logged in, the administration console should look similar to Figure B-12.



Integrated Solutions Console provides a common administrative console for multiple products. The table lists the product suites that can be administered through this installation. Select a product suite to view more information.

Suite Name	Version
WebSphere Application Server	7.0.0.7

Integrated Solutions Console, 7.0.0.7
Build Number: wps0945.05
Build Date: 11/11/09

-
LICENSED MATERIALS PROPERTY OF IBM

Figure B-12 WebSphere Application Server Administration Console

The successful logon shows that the application server is successfully installed and configured for use.

IBM HTTP Server

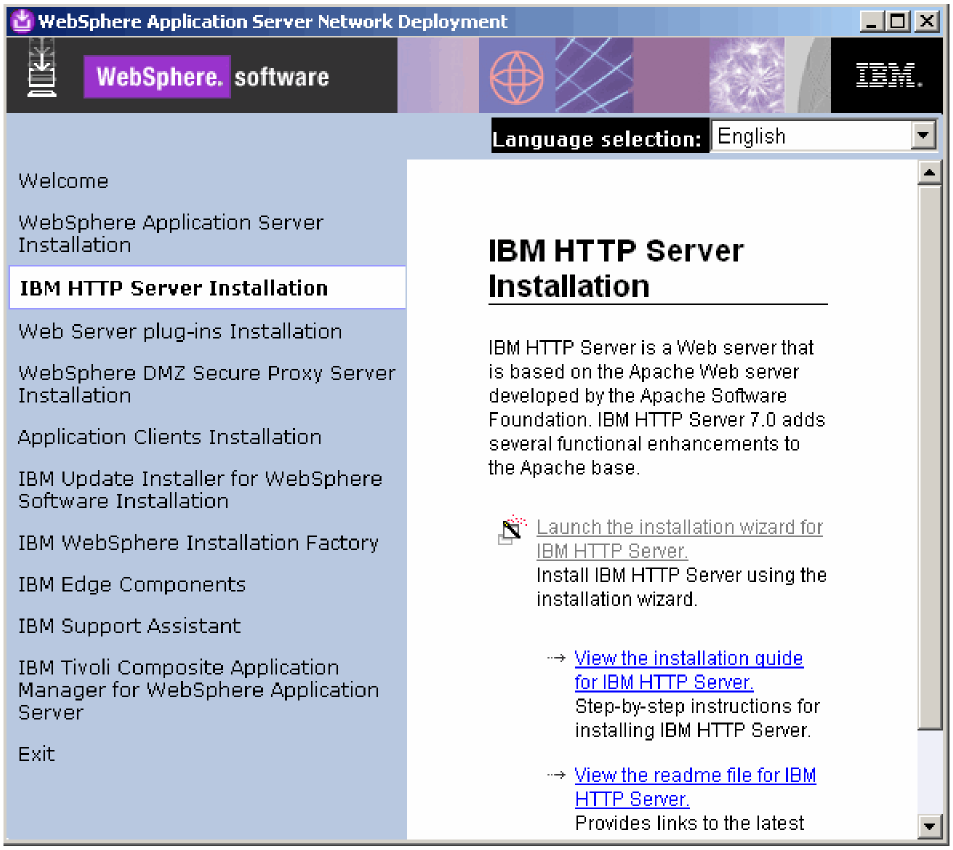
IBM HTTP Server for WebSphere Application Server Version 7.0 is available on the WebSphere Application Server CDs. For detailed instructions on how to install the application server, visit the WebSphere Application Server Version 7.0 Information Center:

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_installihs.html

IBM HTTP Server scenario installation

The following panels show the steps that we took to install the HTTP Server and WebSphere Application Server plug-in on our server in the demilitarized zone.

1. Access the binaries for the WebSphere Application Server supplements and run the launchpad (Figure B-13).


WebSphere Application Server Network Deployment
WebSphere. software
IBM.
Language selection: English
Welcome
WebSphere Application Server Installation
IBM HTTP Server Installation
Web Server plug-ins Installation
WebSphere DMZ Secure Proxy Server Installation
Application Clients Installation
IBM Update Installer for WebSphere Software Installation
IBM WebSphere Installation Factory
IBM Edge Components
IBM Support Assistant
IBM Tivoli Composite Application Manager for WebSphere Application Server
Exit

IBM HTTP Server Installation

IBM HTTP Server is a Web server that is based on the Apache Web server developed by the Apache Software Foundation. IBM HTTP Server 7.0 adds several functional enhancements to the Apache base.

Launch the installation wizard for IBM HTTP Server.
Install IBM HTTP Server using the installation wizard.

View the installation guide for IBM HTTP Server.
Step-by-step instructions for installing IBM HTTP Server.

View the readme file for IBM HTTP Server.
Provides links to the latest

Figure B-13 IBM HTTP Server launchpad menu

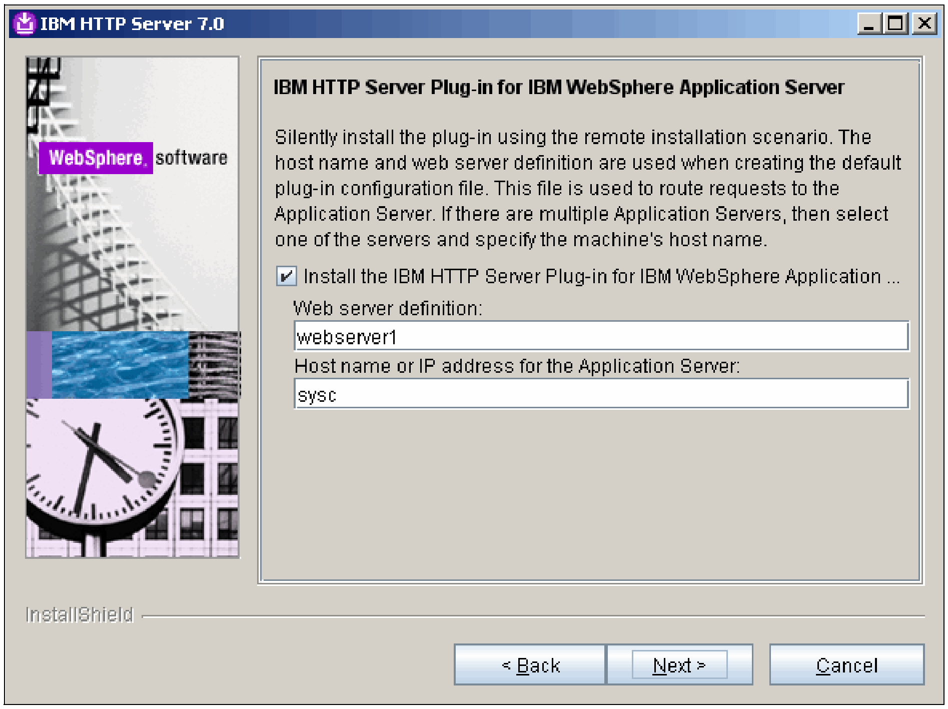
2. Select **IBM HTTP Server Installation** from the left navigational menu and select **Launch Installation Wizard for IBM HTTP Server** from the main navigational menu. When the installation wizard is launched, click **Next**.
3. For the Software License Agreement panel, review the terms, accept them if inclined, and click **Next**.
4. Verify that your system passed the System Prerequisites on the System Prerequisites panel and click **Next**.
5. For the installation location, select where the files should be located. We selected `C:\FTEHTTPServer\`. Click **Next**.
6. For the Port Values Assignment panel, determine the ports available for use and enter the values.

   For our scenario, we set the following:

   – HTTP Port: 81
   – HTTP Administration Port: 8009

   Click **Next**.

7. The Windows Service Definition panel is only encountered on Windows systems. Choose whether the IBM HTTP Server and the IBM HTTP Administration should be run as a Windows service. For our scenario, neither the IBM HTTP Server nor the IBM HTTP Administration are run as a Windows service. Click **Next**.
8. On the HTTP Administration Server Authentication page, enter the user ID and password to administer the HTTP server environment. Click **Next** when complete.
9. Keep the default option to install the HTTP Server Plug-in for the WebSphere Application Server checked. Enter a name for the web server definition and enter your system's host name as shown in Figure B-14. Click **Next**.



IBM HTTP Server Plug-in for IBM WebSphere Application Server

Silently install the plug-in using the remote installation scenario. The host name and web server definition are used when creating the default plug-in configuration file. This file is used to route requests to the Application Server. If there are multiple Application Servers, then select one of the servers and specify the machine's host name.

☑ Install the IBM HTTP Server Plug-in for IBM WebSphere Application ...

Web server definition:
webserver1

Host name or IP address for the Application Server:
sysc

Figure B-14 IBM HTTP Server - IBM HTTP Server plug-in for IBM WebSphere Application Server

10. On the Installation Summary panel, review your choices and select **Next** to begin the installation.
11. When the installation has successfully completed, click **Finish** to exit the installation wizard.
12. To verify the installation, start the IBM HTTP Server and the Administration Server.

For additional information about the ways to start the IBM HTTP Server, see:

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_startihs.html

For more information about ways to start the Administration Server, see the WebSphere Application Server Version 7.0 Information Center:

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_startadmserv.html

For our purposes, we used the menu options to start the IBM HTTP Server and the administration server (Figure B-15).

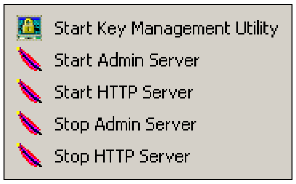


Figure B-15 IBM HTTP Server - Menu Start Options

After the IBM HTTP Server and Administration Server is running, you should be able to access the console through a URL:

```
http://<host>:<HTTP Port>/
```

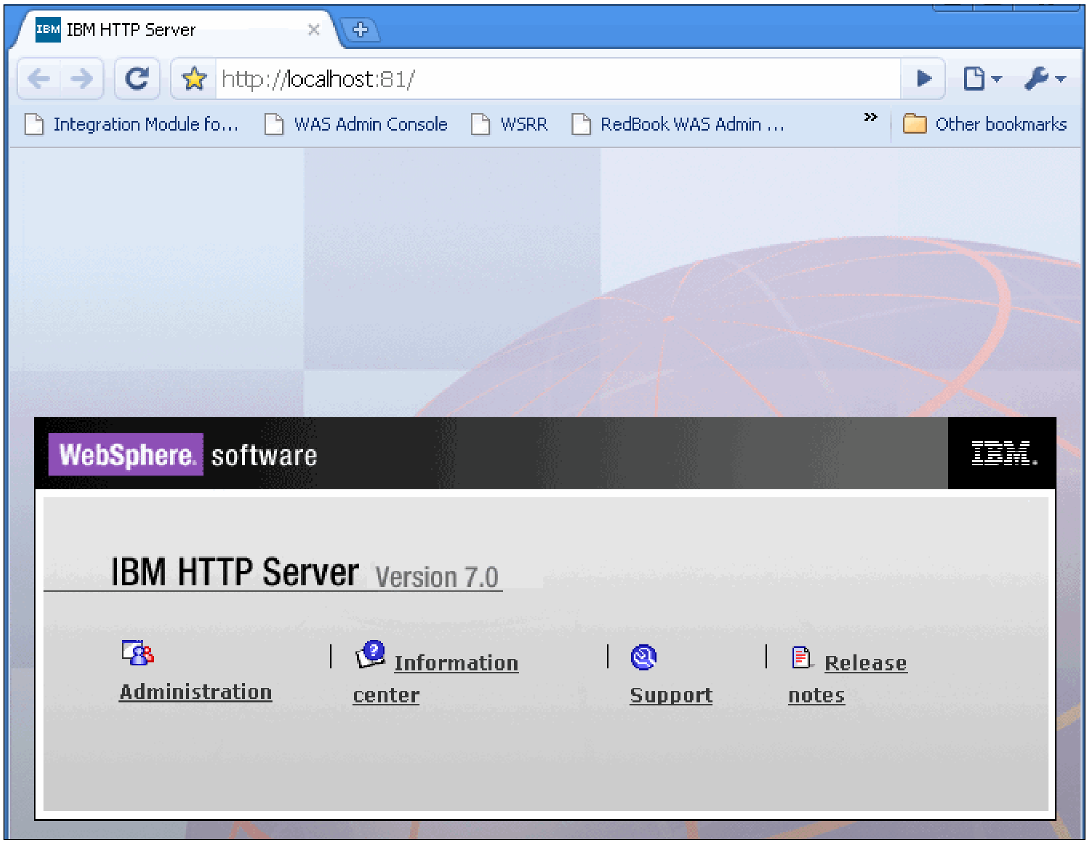


Figure B-16 IBM HTTP Server Administration Console

Inbound communication

Most web applications transmit sensitive data (for example, a user name and password during login or personal data during the interaction with the application). To protect this information during transfer, we use SSL. In the WebSphere environment, application servers are typically accessed through a web server, for example, IBM HTTP Server (IHS).

If client certificate authentication is not required, perform the following steps to configure SSL communication:

1. Configure the web server for Secure Sockets Layer (SSL) (refer to “Configuring the web server for SSL” on page 312):

   a. Create the key database file and certificates required for the web server to participate in an SSL connection. The certificate must be signed by a well-known certificate authority (CA).

   b. Enable the directives in the web server configuration for SSL, pointing to the new key database.

   This step allows SSL connections to be established between web browsers and the web server.

2. Configure the HTTP Plug-in for SSL connections (refer to “Configuring the plug-in for SSL connection” on page 313):

   a. Add the web server definition to WebSphere Application Server.

      When a web server definition is created, it is associated with a keystore that contains all of the signers for the cell and the chained certificate for the web server node. Instructions for creating the web server definition can be found at:

      http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/tihs_remotesetup.html

   b. Copy the web server keystore and stash files for the plug-in to the web server plug-in location.

If client certificate authentication is required, configuration is more complex. In addition to the previous steps, you must configure the web server to require client certificates and configure mutual trust between the plug-in and the application server.

Configuring the web server for SSL

This section illustrates how to implement a secure connection between the browser and the IBM HTTP Web Server. In our example, we use the IBM HTTP Server V7 on a Windows platform. Follow these steps:

1. Create the key database file and certificates that are needed to authenticate with the web server during an SSL handshake.

   In this example, we use a self-signed certificate created by the IBM Key Management Tool (see “IBM Key Management” on page 298). However, for production installation, use a certificate signed by a well known certificate authority.

   Save the key database file as `c:\IBM\HTTPServer\ihskey.kdb`.

2. Enable the SSL directives in the IBM HTTP Server's configuration file (`httpd.conf`):

   a. Remove the comment from the ibm_ssl_module to load this module:

   ```
   LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
   ```

   b. Create an SSL VirtualHost stanza using the following example and directives:

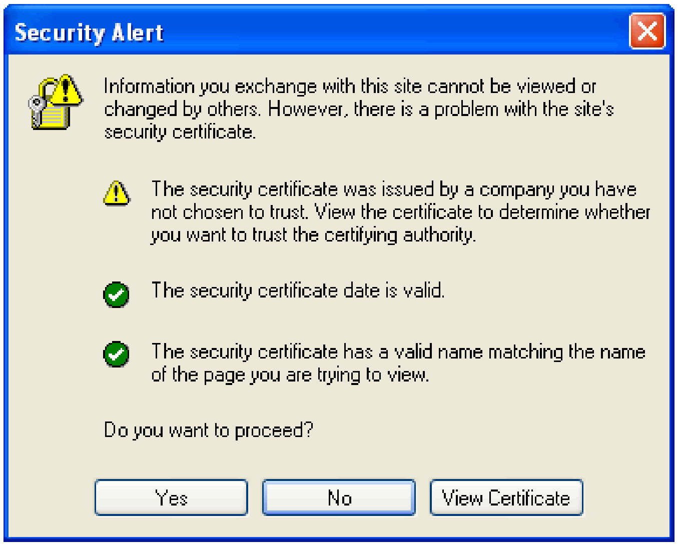
   ```
   Listen 443
   <VirtualHost  *:443>
   SSLEnable
   SSLClientAuth None
   </VirtualHost>
   SSLDisable
   KeyFile "c:/IBM/HTTPServer/ihskey.kdb"
   SSLV2Timeout 100
   SSLV3Timeout 1000
   ```

3. Save the configuration file and restart the IBM HTTP Server.

4. Access the IBM HTTP Server welcome page from a web browser:

   ```
   https://your-host name
   ```

   You might see a Security Alert page (Figure B-17).



Security Alert

Information you exchange with this site cannot be viewed or changed by others. However, there is a problem with the site's security certificate.

The security certificate was issued by a company you have not chosen to trust. View the certificate to determine whether you want to trust the certifying authority.

The security certificate date is valid.

The security certificate has a valid name matching the name of the page you are trying to view.

Do you want to proceed?

Figure B-17 SSL Security Alert

   This message occurs because the configuration uses a self-signed certificate that is not issued by a trust-certifying authority.

5. Click **Yes**. You will see the welcome page.

The next step is to secure the communication between the plug-in and application server with SSL.

Configuring the plug-in for SSL connection

This section illustrates how to configure the HTTP plug-in to enable SSL connections to the application server.

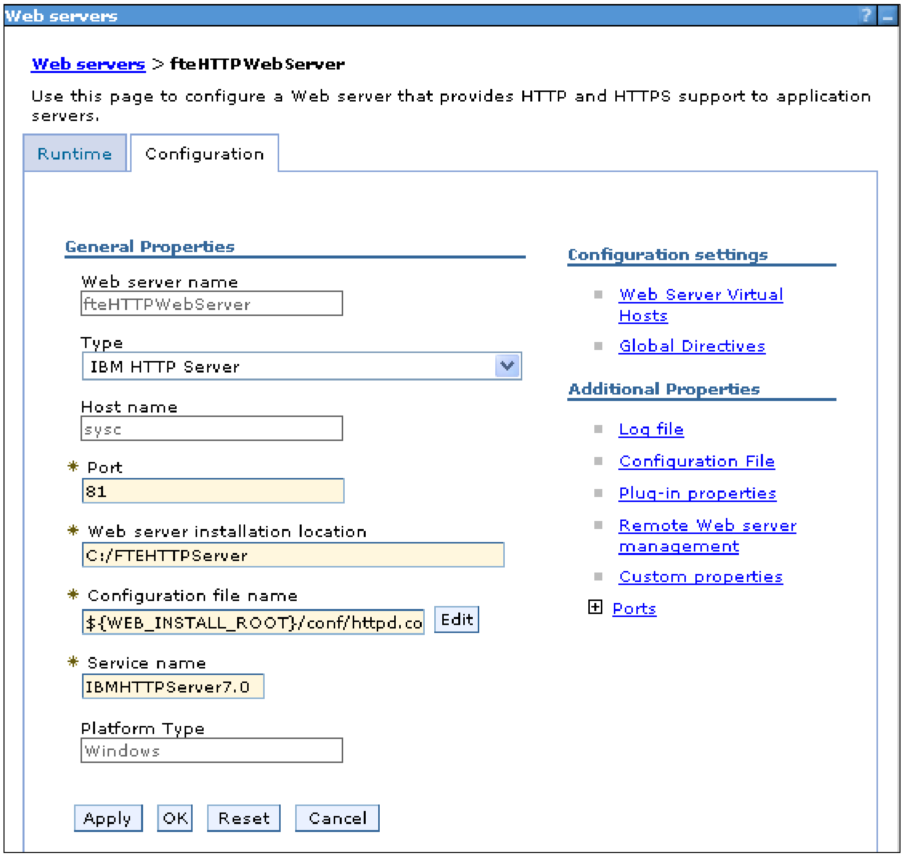
In this example, we assume that a definition for the web server has been created in WebSphere Application Server. When the web server definition was created, WebSphere Application Server associated the web server plug-in with a Certificate Management Services

(CMS) keystore for a specific node. This keystore contains all of the signers for the cell with the self-signed or chained certificate for the node. The plug-in can communicate securely to WebSphere Application Server, even when the plug-in is configured with SSL client authentication enabled, because its personal certificate is signed by the default root certificate, which is trusted by the application server.

For directions on how to configure the web server definition, see the section “Setting up a remote web server” at the WebSphere Application Server version 7.0 Information Center:

`http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/tihs_remotesetup.html`

Our web server was configured according to the directions in the WebSphere Application Server Version 7.0 Information Center. Figure B-18 shows our web server properties.



Web servers > fteHTTPWebServer

Use this page to configure a Web server that provides HTTP and HTTPS support to application servers.

General Properties

Web server name: fteHTTPWebServer

Type: IBM HTTP Server

Host name: sysc

* Port: 81

* Web server installation location: C:/FTEHTTPServer

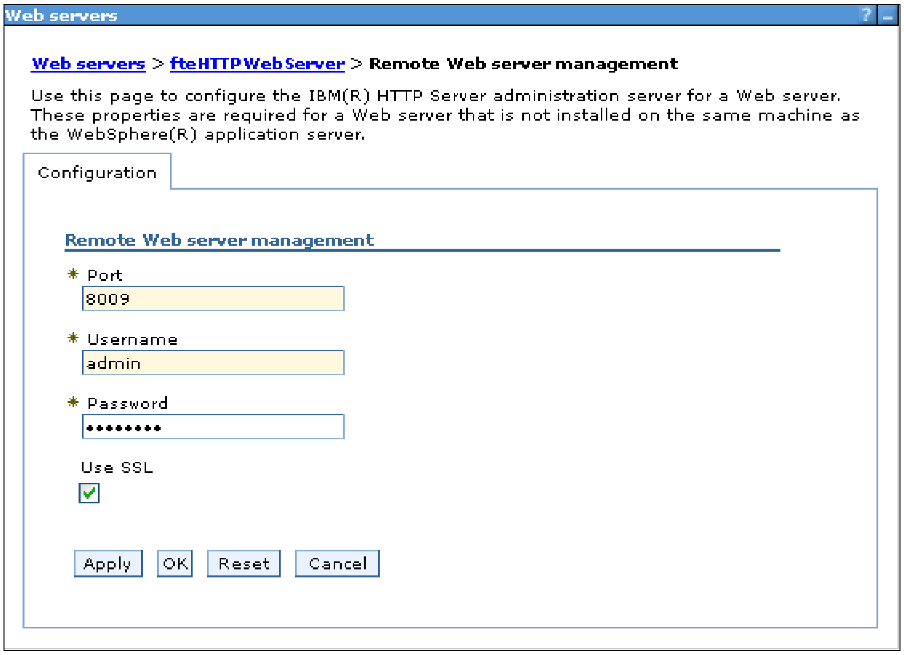
* Configuration file name: ${WEB_INSTALL_ROOT}/conf/httpd.co

* Service name: IBMHTTPServer7.0

Platform Type: Windows

Figure B-18 Web server properties

Figure B-19 shows the remote web server management properties.



Web servers > fteHTTPWebServer > Remote Web server management

Use this page to configure the IBM(R) HTTP Server administration server for a Web server. These properties are required for a Web server that is not installed on the same machine as the WebSphere(R) application server.

Remote Web server management

* Port
8009

* Username
admin

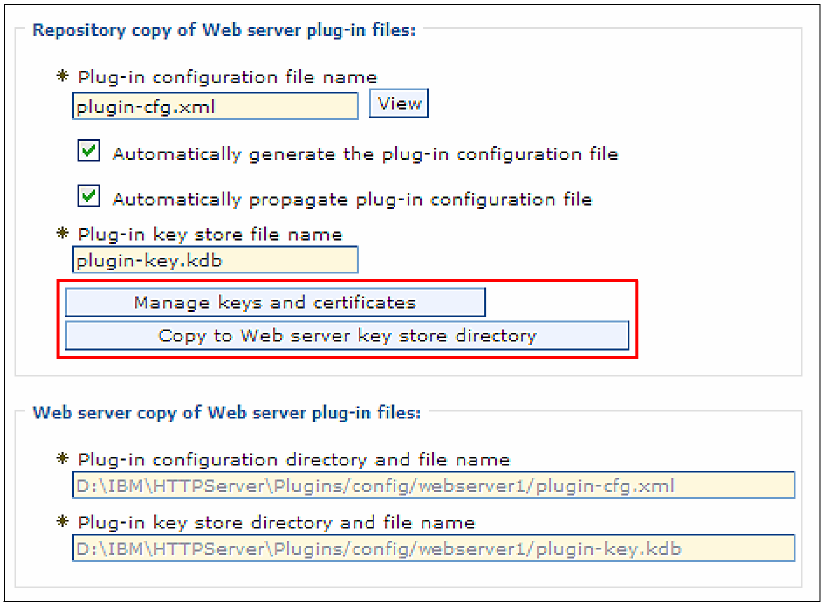
* Password
••••••••

Use SSL
☑

Figure B-19 Remote web server management

Complete this configuration by performing the following steps:

1. From the WebSphere administrative console, click **Servers** → **Server Types** → **Web servers**.
2. Click the web server name to open the configuration page.
3. Click **Plug-in properties**.
4. Click **Manage keys and certificates** to see the plug-in keystore details (CMSKeyStore) (Figure B-20).


Repository copy of Web server plug-in files:

* Plug-in configuration file name
plugin-cfg.xml

☑ Automatically generate the plug-in configuration file

☑ Automatically propagate plug-in configuration file

* Plug-in key store file name
plugin-key.kdb

Manage keys and certificates
Copy to Web server key store directory

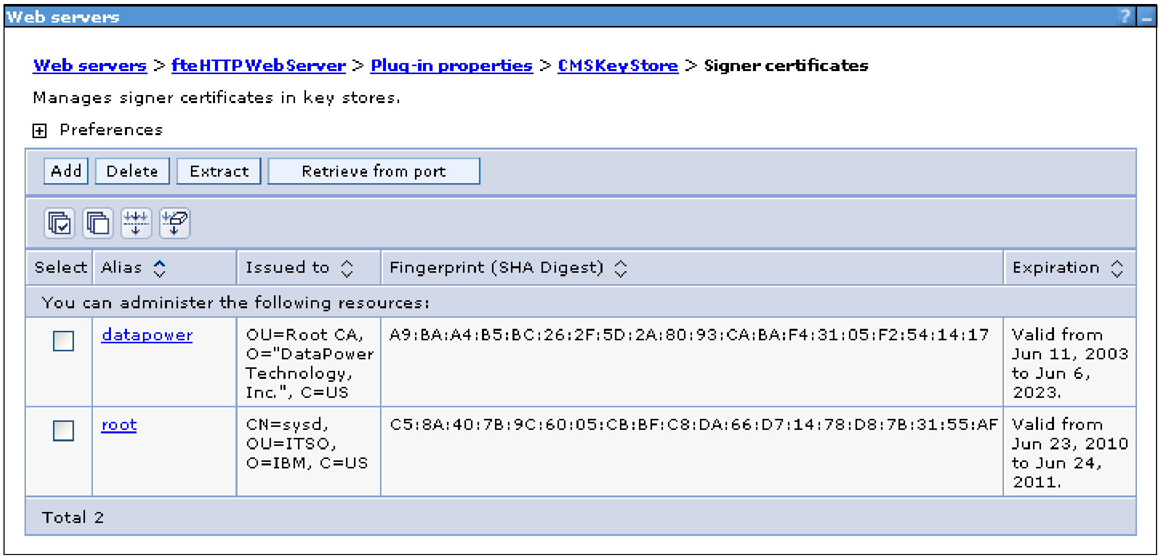
Web server copy of Web server plug-in files:

* Plug-in configuration directory and file name
D:\IBM\HTTPServer\Plugins/config/webserver1/plugin-cfg.xml

* Plug-in key store directory and file name
D:\IBM\HTTPServer\Plugins/config/webserver1/plugin-key.kdb

Figure B-20 Plug-in properties

5. Click **Signer certificates** to see a list of the signer certificates. We have to ensure that all nodes’ root certificates are there (Figure B-21).



Web servers > fteHTTPWebServer > Plug-in properties > CMSKeyStore > Signer certificates

Manages signer certificates in key stores.

Select	Alias	Issued to	Fingerprint (SHA Digest)	Expiration
You can administer the following resources:				
□	datapower	OU=Root CA, O="DataPower Technology, Inc.", C=US	A9:BA:A4:B5:BC:26:2F:5D:2A:80:93:CA:BA:F4:31:05:F2:54:14:17	Valid from Jun 11, 2003 to Jun 6, 2023.
□	root	CN=sysd, OU=ITSO, O=IBM, C=US	C5:8A:40:7B:9C:60:05:CB:BF:C8:DA:66:D7:14:78:D8:7B:31:55:AF	Valid from Jun 23, 2010 to Jun 24, 2011.
Total 2				

Figure B-21 List of signer certificates

The default root certificate and the node root certificate are automatically added to the keystore. If you add other nodes later, their root certificates will be added to this keystore. However, the keystore is not automatically propagated to web servers.

6. Copy the web server keystore and stash files to the web server machine:

   a. If you use managed web servers, click **Copy to Web server keystore directory** on the plug-in properties page (Figure B-20 on page 315).

      When you use this option, the plug-in keyring file is propagated from `*node_profile_root*/servers/*web_server_def*/plugin-key.kdb` on the deployment manager system to `*plug-in_root* /config/*web_server_def*/plugin-key.kdb` on the web server computer.

      If you use an unmanaged web server, manually copy the keystore and stash files to the web server.

   b. Ensure that the plug-in configuration points to the location where you stored the files. The following transport directives in the `plug-in.xml` configuration file contain the location:

```
<Transport Hostname="sysd" Port="9444" Protocol="https">

<Property Name="keyring"
Value="c:\HTTPServer\Plugins\config\fteHTTPWebServer\plugin-key.kdb"/>

<Property Name="stashfile"
Value="c:\HTTPServer\Plugins\config\fteHTTPWebServer\plugin-key.sth"/>

</Transport>
```

      Note that these directives will not be included in the `plug-in.xml` file until you have associated the web server with an application module.

7. Ensure that the virtual host that is mapped to the applications includes the SSL port and that the plug-in file is refreshed with the latest configuration. You might have to add the HTTP port and the HTTP Administration port for the HTTP server to the virtual host.

C

Additional material

This book refers to additional material that can be downloaded from the internet as described below.

Locating the web material

The web material associated with this book is available in softcopy on the internet from the IBM Redbooks web server. Point your web browser at:

`ftp://www.redbooks.ibm.com/redbooks/SG247886`

Alternatively, you can go to the IBM Redbooks website at:

ibm.com/redbooks

Select **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG247886.

Using the web material

The additional web material that accompanies this book includes the following files:

Folder name	***Description***
`Common_Files`	Files to support the configuration and testing of the scenarios found in Chapter 7, “B2B-enabled managed file transfer” on page 109, and Chapter 8, “Integrating partner transfers with internal ESB” on page 175
`B2BScenario_Files`	Files to support the configuration and testing of the scenario found in Chapter 7, “B2B-enabled managed file transfer” on page 109
`BrokerScenario_Files`	Files to support the configuration and testing of the scenario found in Chapter 8, “Integrating partner transfers with internal ESB” on page 175

How to use the web material

This web material can be used to build the scenarios found in Chapter 7, “B2B-enabled managed file transfer” on page 109, and Chapter 8, “Integrating partner transfers with internal ESB” on page 175. To use the material for either scenario, you must have the prerequisites listed for each.

To use this material, create a subdirectory (folder) on the local drive of the desktops being used to access the XB60 and WebSphere Message Broker Toolkit, and uncompress the contents of the web material compressed file into this folder. These files are referenced as you go through the scenario.

Common_Files folder

Table 8-8 lists and describes the files in the `Common_Files` folder.

Table 8-8 Files in the Common_Files directory

Type	Directories and files	Description
File	`generate-mqfte-request.xsl`	XSLT file used in the multi-protocol gateway that integrates the XB60 with WebSphere MQ File Transfer Edition.
File	`Partner_Domain.zip`	Export of the PARTNER domain used in the XB60 to simulate the external trading partner's hub. For information about importing domains to your XB60, see: http://publib.boulder.ibm.com/infocenter/wsdatap/v3r8m1/topic/xb60/administratorsguide120.htm#importconfig_importingconfigurationdata_task

BrokerScenario_Files folder

Table 8-9 lists and describes the file in the `BrokerScenario_Files` folder.

Table 8-9 Files in the BrokerScenario_Files directory

Type	Directories and files	Description
File	`StkReplReq.xml`	XML input file used to test the scenario.
File	`XB60FTEWMBFlowProject.zip`	Compressed file containing the WebSphere Message Broker project.This file was exported from a WebSphere Message Broker Toolkit V7.0.0.1 workspace. This file can be imported into the WebSphere Message Broker Toolkit. In the toolkit, select **File** → **Import** → **Other** → **Project Interchange**. Click **Next** and click **Browse** to select the compressed file. On the Import Projects panel, check the box by XB60FTEWMBFlowProject and click **Finish**.

B2BScenario_Files folder

Table 8-10 and Table 8-10 list and describe the files in the BrokerScenario_Files folder.

Table 8-10 Files in the Cert directory

Type	Directories and files	Description
Dir	`Cert`	The Cert directory houses certificates used in the B2B-enabled managed file transfer scenario
File	`b2bfte-privkey.pem`	The private key to be used in the B2BFTE internal profile for AS signatures and decryption
File	`b2bfte-sscert.pem`	The public certificate paired with the private key to be used in the B2BFTE internal profile for signatures and decryption and to be provided to the external trading partner for AS encryption
File	`partner-privkey.pem`	The private key to be used for AS signatures and decryption in the PARTNER internal profile in the PARTNER domain being used to simulate the external partner
File	`partner-sscert.pem`	The public certificate paired with the private key to be used in the PARTNER internal profile for signatures and decryption and to be provided to the B2BFTE domain for AS encryption

Table 8-11 Files in the sample_files directory

Type	Directories and files	Description
Dir	`sample_files`	The `sample_files` directory houses the files used in the B2B-enabled managed file transfer scenario
File	`b2bfte_partner.edi`	The sample file used to test the B2B-enabled managed file transfer scenario outbound flow
File	`partner_b2bfte.edi`	The sample file used to test the B2B-enabled managed file transfer scenario inbound flow

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks publications

For information about ordering these publications, see "How to get Redbooks" on page 322. Note that certain documents referenced here might be available in softcopy only.

- *B2B Appliances: Creating Customer Value Through Exceptional B2B Messaging Performance and Security*, REDP-4524
- *WebSphere Application Server V7.0 Security Guide*, SG24-7660
- *IBM WebSphere DataPower B2B Appliance XB60 Revealed*, SG24-7745
- *Getting Started with WebSphere MQ File Transfer Edition V7*, SG24-7760
- *Managed File Transfer for SOA using IBM WebSphere MQ File Transfer Edition*, REDP-4533

Online resources

These websites are also relevant as further information sources:

- WebSphere MQ File Transfer Edition

  http://www-01.ibm.com/software/integration/wmq/filetransfer/
- WebSphere MQ File Transfer Edition System Requirements

  http://www-01.ibm.com/software/integration/wmq/filetransfer/requirements/
- FO02: WebSphere MQ File Transfer Edition - Web Gateway

  http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24026419&loc=en_US&cs=utf-8&lang=en
- WebSphere MQ File Transfer Edition Version 7.0 Information Center

  http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/index.jsp
- WebSphere Message Broker

  http://www-01.ibm.com/software/integration/wbimessagebroker/
- WebSphere Message Broker System Requirements

  http://www-01.ibm.com/software/integration/wbimessagebroker/requirements/
- WebSphere Message Broker Version 7.0 Information Center

  http://publib.boulder.ibm.com/infocenter/wmbhelp/v7r0m0/index.jsp
- WebSphere MQ Version 7.0 Information Center

  http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp
- IBM DataPower B2B Appliance XB60 Information Center

  http://publib.boulder.ibm.com/infocenter/wsdatap/v3r8m1/index.jsp

- WebSphere Adapters

  `http://www-01.ibm.com/software/integration/wbiadapters/`
- WebSphere Application Server Version 7.0 Information Center

  `http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp`
- WebSphere for z/OS Version 7 - Configuration Planning Spreadsheets

  `http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS3341`

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

IBM

Redbooks

Multi-Enterprise File Transfer with WebSphere Connectivity

Multi-Enterprise File Transfer with WebSphere Connectivity

Enable end-to-end file transfers using WebSphere MQ File Transfer Edition

Externalize your file transfers with the DataPower B2B appliance

Enhance file transfer with WebSphere Message Broker

This IBM Redbooks publication describes how to exchange data between applications running in two different enterprises reliably and securely. This book includes an overview of the concepts of managed file transfer, the technologies that can be used, and common topologies for file transfer solutions. It then provides four scenarios that address different requirements. These scenarios provide a range of options that can be suited to your individual needs. This book is intended for anyone who needs to design or develop a file transfer solution for his enterprise.

The first scenario shows the use of an HTTPS web gateway to allow files to be transferred from an external web client to an internal WebSphere MQ File Transfer Edition backbone network. This option uses the WebSphere MQ File Transfer Edition Web Gateway SupportPac FO02.

The second scenario uses the WebSphere MQ File Transfer Edition bridge agent to allow files to be transferred from an external FTP/SFTP server to a WebSphere MQ File Transfer Edition backbone network.

The third scenario extends the concept of file transfer between enterprises by introducing more sophisticated transfer capabilities, along with enhanced security. This scenario uses the IBM DataPower B2B Appliance XB60 to look at the specific case of file transfers between business partners.

The last scenario also illustrates the integration of the IBM DataPower B2B Appliance XB60 and WebSphere MQ File Transfer Edition, but in this case, non-B2B protocols are used. The file transfer is further enhanced through the use of WebSphere® Message Broker to mediate the file transfer for routing and protocol transformation within the enterprise.

SG24-7886-00 ISBN 0738434728